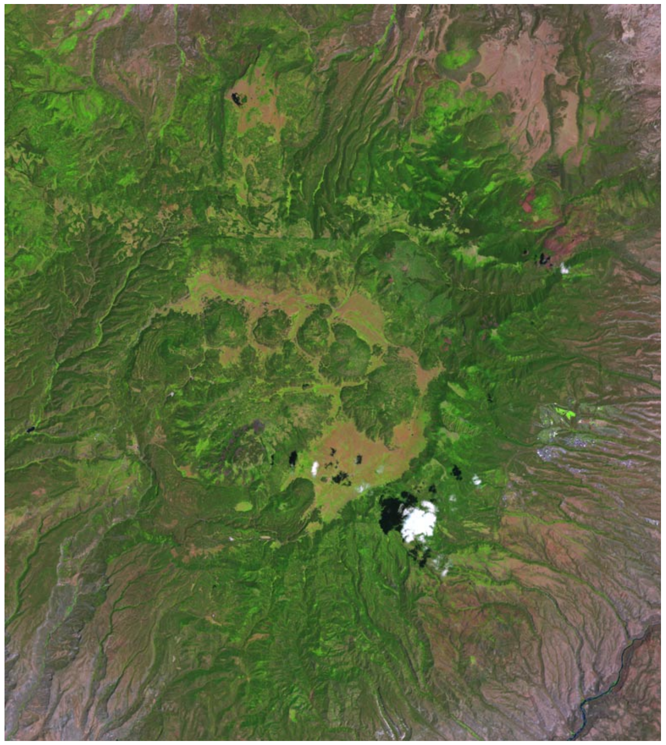
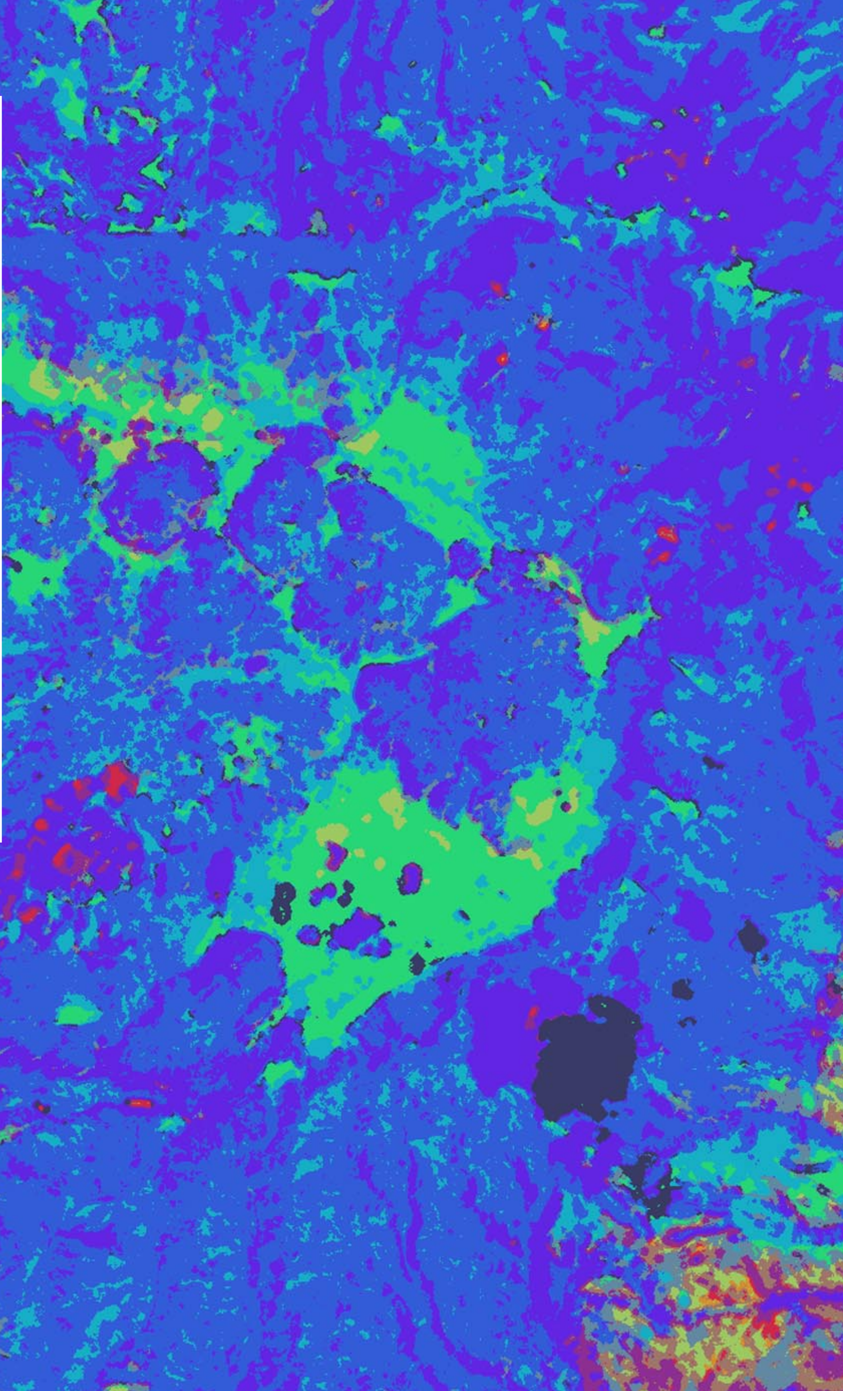
2002 WINNER R&D 100

2002 R&D 100 Entry

GENIE
Evolving Feature-Extraction Algorithms for Image Analysis

The GENIE Team





Evolves software for spatial and spectral image processing

Addresses diverse problems, from natural disasters to medical diagnosis

Easy-to-use graphical interface

Parallel processing using off-the-shelf computer hardware

Los Alamos
NATIONAL LABORATORY

2002 R&D 100 Entry

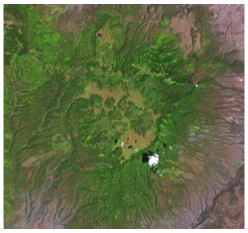
GENIE
Evolving Feature-Extraction Algorithms for Image Analysis

The GENIE Team

ABOUT THE COVER:

The inset photo, a satellite image of the area affected by the Cerro Grande wildfire that occurred near Los Alamos in May 2000, contains seven spectral bands of imaging information. Analyzing the image would normally require considerable effort by scientists working closely with computer programmers. In this case, a single image analyst trained GENIE to identify land-cover classes in one session. The main photo shows the map GENIE produced for 12 land-cover classes. Forest classes (pine, conifers, aspen) appear in shades of blue, grassland in green, and bare ground and pinon-juniper scrublands appear in shades of red and brown. The algorithm that GENIE generated to produce this map can also be used to identify land-cover classes in other images.

Los Alamos
NATIONAL LABORATORY



Executive Summary

GENIE: Evolving Feature-Extraction Algorithms for Image Analysis

Features

GENIE (GENetic Imagery Exploitation) mimics evolution in order to create more-effective algorithms for detecting features in digital images produced by a variety of remote-sensing techniques. GENIE assembles an initial set of low-level image-processing algorithms (e.g., edge detectors, texture measures, and spectral operators) and then tests each algorithm's ability to find the feature of interest. The "less fit" algorithms are discarded; the "more fit" ones are combined to produce superior ones. After several generations of survival of the fittest, the resulting algorithm is highly optimized. Although features and imagery constantly change, GENIE's ability to evolve superior algorithms allows it to find the features of interest in nearly any set of images.

Applications

GENIE can be used to

- map damage caused by wildfires, snowstorms, tornados, hurricanes, floods, tsunamis, earthquakes, volcanoes, or terrorist attacks,
- monitor environmental changes or crop health,
- track population growth,
- detect signs of disease in medical images,
- detect defects in products made in assembly lines,
- detect weapons and explosives at airport security checkpoints,
- detect suspect vehicles in traffic, and
- create maps of craters, canyons, and plateaus on other planets to assist in choosing landing sites.

Benefits

Analysts with no programming experience can quickly learn how to use GENIE to evolve algorithms that extract specific features from digital images.

GENIE's results can be reused to help GENIE build up its "understanding" of complex tasks. (For example, after GENIE learns to find water, it then can easily learn to find beaches.)

GENIE learns to ignore unimportant image-to-image variations such as atmospheric haze or variations in overall illumination.

2002 R&D 100 Award Entry Form

1. Submitting organization

Organization	Los Alamos National Laboratory
Address	Mail Stop D436
City, State, ZIP	Los Alamos, NM 87545
Country	USA
Submitter	Steven P. Brumby
Phone	(505) 667-3768
Fax	(505) 665-4414
E-mail	brumby@lanl.gov

AFFIRMATION: I affirm that all information submitted as a part of, or supplemental to, this entry is a fair and accurate representation of this product.

(Signature) SP Brumby

2. Joint submitter(s)

N/A

3. Product name

GENIE: Evolving Feature-Extraction Algorithms for Image Analysis

4. Brief product description

GENIE uses genetic programming to generate algorithms that are customized to find specific features in digital images from remote-sensing instruments.

5. Eligibility: When was this product first marketed or available for order?

Month	June
Year	2001

6. Principal developer(s)

Name	Nancy A. David
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Address	MS D436
City, Sate, ZIP	Los Alamos, NM 87545
Country	USA
Phone	(505) 667-8896
Fax	(505) 665-4414
E-mail	ndavid@lanl.gov

7. Product price

GENIE will cost approximately $3,000 per software package.

8. Do you hold any patents or patents pending on this product?

NO

9. Primary function

Libraries of multispectral data are being produced by satellite and airborne remote-sensing instruments. To exploit the information they contain on a large scale requires automated extraction of the features of interest—whether they are vegetation patterns, bodies of water, buildings, or roads. The sheer volume of data accumulating from a burgeoning variety of remote-sensing techniques defies human analysis.

In the past, attempts to automate feature-identification tasks have produced algorithms (computerized procedures) that detect features against common image backgrounds. But these algorithms are expensive to develop, require scientists who can write computer code, and are often so specialized that new algorithms must be refined or generated for each new image background, which can vary, for example, with the season.

We have developed a highly versatile software package that refines and automates the algorithm-generation process itself. GENIE (GENetic Imagery Exploitation) mimics evolution to create algorithms that detect specific features against image backgrounds that are often complex and typically consist of millions of pixels. GENIE uses genetic programming to evolve these image-processing algorithms but goes well beyond genetic programming's usual level of complexity. (The paper "GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multi-Spectral Images," in the Appendix describes how GENIE creates and uses these algorithms.)

GENIE initially assembles a random population of image-processing algorithms and then, with the help of a human analyst, tests each algorithm's ability to find the feature of interest—for example, crop patterns or complex terrain. The "less fit" algorithms are discarded, and the "more fit" ones combined to produce superior algorithms. After many generations, an ideal algorithm evolves that can identify the specified feature in the image on which GENIE was trained. But GENIE can apply what it has learned to quickly tailor other algorithms to find the same feature in other images.

With GENIE, a human analyst can also develop algorithms that analyze images obtained in several spectral bands (visible, thermal-infrared, or ultraviolet light, for example), from various types of sensors (such as laser altimeters or radar), and at various times. GENIE is particularly effective at searching for complex features that contain both spatial (shape) and spectral (wavelength) identifiers, such as wildfire burn scars, cancer cells, and even golf courses (see Figures 1–7 in the Appendix). GENIE can also sequentially extract many features from the same image to produce, for example, land-cover classifications (which identify various types of vegetation), such as that on the cover of this entry.

GENIE runs on clusters of standard UNIX workstations and typically takes a few hours to evolve a custom algorithm, which can be applied to new images in seconds. GENIE produces algorithms that are robust, which means they can reliably identify features despite background variations caused by atmospheric haze or changes in overall scene illumination. GENIE can also generalize well beyond conventional, purely spectral or purely spatial classification algorithms, as has been demonstrated in a published study (see "Comparison of GENIE and Conventional Supervised Classifiers for Multispectral Image Feature Extraction" in the Appendix).

A videotape accompanying this entry shows how GENIE works.

10A. Competitors

Three supervised classification techniques are commonly used in software that analyzes remote-sensing images: maximum likelihood, Mahalanobis distance, and spectral angle mapper. We include the maximum likelihood classifier as one of our competitors because it is the most widely used technique. Our second competitor is the traditional approach: human analysts who interpret photos manually or use computer codes written for specific image-analysis tasks.

10B. Comparison matrix

Parameters	GENIE	Maximum Likelihood Classifier	Human Analyst	Comments
Spatial and spectral data used	Yes	Spectral only	Mostly spatial, limited spectral	Techniques using only spectral or spatial image data have limited abilities to identify features or to discriminate against unimportant variations in image background. GENIE’s algorithms provide superior performance by using both types of information.
Uses images from many sources	Yes	Yes, but only spectral	Limited	Automated techniques such as GENIE and maximum likelihood classifiers can process images from several sources, but a human analyst can easily be overwhelmed when trying to understand and combine data from multiple sources.
Background discrimination	Yes	Limited	Limited	Raw data contains image-to-image variations in illumination that can interfere with identifying features. These variations are caused by atmospheric haze, the position of the sun (which varies both seasonally and daily), seasonally dependent background (autumn leaves, for example), and aging effects on the imaging instruments. Analysis of the data must discriminate between these background variations and those that characterize relevant features. Subjectivity and fatigue plague human analysts, poorly equipping them to discriminate between these variations. Because maximum likelihood classifiers process only the raw image data, they also cannot discriminate against them. GENIE’s use of spatial and spectral image data allows it to identify and discount background variations.
Analysis speed	100 times faster than human analysis	Faster than human analysis but not robust	Slow	GENIE’s algorithms can find impermeable surfaces (finished roads and roofs) at least 100 times faster than a human analyst can. In addition to producing a superior algorithm to identify the feature of interest, GENIE also produces a map of the feature during the training session so the analyst can assess GENIE’s performance. Further, the algorithm can be immediately used to extract the same feature from other images, with additional dramatic time savings. Maximum likelihood classifiers, which use only spectral information, can be faster to apply than GENIE but often fail to find the same feature in similar images collected under slightly different illumination or weather conditions.
Knowledge building	Finds spatio-spectral signatures	Limited to spectral signatures	Qualitative only	Because GENIE’s algorithms are written in a human-readable programming language (IDL), they can be easily understood and used to determine the distinguishing spatial and spectral attributes (the spatio-spectral signature) of the feature of interest. Manual techniques provide only qualitative information. Maximum likelihood classifiers can provide information only on spectral signatures.

10C. Improvements on competitive technologies

GENIE creates algorithms quickly. In the past, highly trained professionals wrote code for image-processing algorithms customized to identify each feature of interest from its spatial and/or spectral characteristics for a specific background—a time-consuming and expensive task. With GENIE, analysts can cost-effectively develop customized algorithms in minutes to hours instead of the weeks to months formerly required.

GENIE algorithm development is user friendly. A human analyst with no previous programming experience can immediately use GENIE to create customized algorithms that process spatial and spectral data in images produced by a variety of remote-sensing instruments. The analyst can quickly see where GENIE is having trouble and adjust the training data to help GENIE zero in on the feature of interest.

GENIE algorithms discriminate against unimportant image-to-image background variations. Because GENIE combines a variety of low-level image-processing algorithms, it can easily find the characteristic spatial and spectral signatures of the features of interest. This combination of cues increases the likelihood of successfully evolving algorithms that can discriminate against unimportant background variations, such as clouds, atmospheric haze, and varying overall illumination. Such robustness is a dramatic improvement over GENIE's competitors. (The journal preprint "Comparison of GENIE and Conventional Supervised Classifiers for Multispectral Image Feature Extraction" in the Appendix discusses experiments involving comparisons between GENIE and several supervised classification techniques for a number of classification tasks that use multispectral, remotely sensed imagery.)

GENIE is compatible with existing image analysis systems. GENIE launches from standard image-analysis software packages, which are designed to handle mundane image-file handling (reading/writing in instrument-specific formats) and mapping details (such as coregistering datasets and changing geographical projection), and GENIE's results can later be imported by the standard software for further analysis. Results also can be easily reused to help GENIE acquire knowledge of complex tasks.

11A. Principal applications

GENIE was originally designed to analyze images taken from satellites and aircraft for several related applications:

- Monitoring environmental changes in complex ecosystems (for example, by providing detailed maps of erosion or the locations of vegetation and water—as shown in Figure 3 in the Appendix),

- Mapping the progress or consequences of wildfires, hurricanes, tornadoes, snowstorms, floods, tsunamis, earthquakes, and volcanoes. (See the article "Evolving forest fire burn severity classification algorithms for multi-spectral imagery" in the Appendix. Figure 4 in the Appendix shows GENIE's map of the damage caused by the intense Cerro Grande wildfire.)
- Creating detailed land-use maps for monitoring trends such as urban growth and masks of background features such as clouds so that they can be removed from further analysis, as shown in Figure 2 in the Appendix.

11B. Other applications

GENIE can be used to enhance a variety of imaging tasks:

- Noninvasively searching for weapons and explosives at security checkpoints in schools, government buildings, and airports (see Figure 5 in the Appendix).
- Mapping damage caused by terrorists—for example, the terrorist attacks in New York City.
- Screening medical images, such as x-rays or microscope images used to detect bacteria or cancerous cells (see Figure 7 in the Appendix).
- Assessing the health of commercial crops.
- Inspecting for product defects on the assembly line.
- Creating maps of other planets to identify the craters, surface minerals, and erosion features such as canyons that are crucial to understanding the past and present state of our neighbors in the solar system and for selecting suitable landing sites (see Figure 6 the Appendix).

12. Summary

The volume and variety of digital images now available to scientists are truly staggering. They include Hubble Space Telescope images of the most distant objects in the universe; aircraft and satellite images of the entire surface of Earth taken in various spectral bands; images taken by NASA's lunar and Martian orbiters and rovers; spacecraft images of other planets in the solar system as well as of asteroids; and medical images such as conventional x-rays, biopsy micrographs, and CAT, MRI, PET, and ultrasound images. All of these images contain information that until now could not be fully interpreted because of the limitations of human image analysis and of existing image-analysis software. Because it automates the algorithm-development process, GENIE is an important image-analysis tool, making possible the tasks and programs that are unthinkable with traditional manual analysis or existing image-analysis software.

To choose an example uncomfortably close to home, within hours after the May 2000 Cerro Grande wildfire wreaked havoc near and in Los Alamos, New Mexico, Los Alamos scientists used GENIE to provide detailed maps to support environmental assessment and restoration efforts. The fire consumed more than 43,000 acres of forest and destroyed 235 homes. GENIE generated updated, consistent vegetation maps for an area covering approximately 1.2 million acres, which had been previously imaged only by a patchwork of maps manually produced over the years by many federal, state, and local agencies. Without GENIE, such timely, detailed assessment of fire damage would have been impossible.

GENIE will never replace the highly skilled scientists or photo-interpretation analysts that develop and apply remote-sensing technology. However, the accurate, robust, automated feature-extraction tools it evolves will play an important part in enabling such experts to exploit the wealth of data being transmitted by imaging sensors around the globe.

Organization Data

13. Chief executive officer

Name	Dr. John C. Browne
Position	Director
Organization	Los Alamos National Laboratory
Address	Mail Stop A100
City, State, ZIP	Los Alamos, NM 87545
Country	USA
Phone	(505) 667-5101
Fax	(505) 667-2997
E-mail	browne@lanl.gov

14. Contact person to handle all arrangements on exhibits, banquet, and publicity

Name	Cindy Boone
Position	R&D 100 Coordinator
Organization	Los Alamos National Laboratory
Address	Mail Stop C333
City, State, ZIP	Los Alamos, NM 87545
Country	USA
Phone	(505) 667-1229
Fax	(505) 665-3125
E-mail	boone@lanl.gov

15. To whom should reader inquiries about your product be directed?

Name	Nance A. David
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Address	Mail Stop D436
City, State, ZIP	Los Alamos, NM 87545
Country	USA
Phone	(505) 667-8896
Fax	(505) 665-4414
E-mail	ndavid@lanl.gov

Appendix

Six copies of a videotape showing how GENIE works accompany this entry.

List of Co-developers

Name	Jeffrey J. Bloch
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 665-2568
Fax	(505) 667-3815
E-mail	jbloch@lanl.gov

Name	Steven P. Brumby
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 667-3768
Fax	(505) 665-4414
E-mail	brumby@lanl.gov

Name	Nancy A. David
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 667-8896
Fax	(505) 665-4414
E-mail	ndavid@lanl.gov

Name	Diana Esch-Mosher
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 665-0012
Fax	(505) 665-4197
E-mail	desch-mosher@lanl.gov

Name	Mark Galassi
Position	Contractor
Organization	Los Alamos National Laboratory
Phone	(505) 665-2030
Fax	(505) 665-4414
E-mail	mgalassi@lanl.gov

Name	Neal R. Harvey
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 667-9077
Fax	(505) 665-4414
E-mail	harve@lanl.gov

Name	Simon Perkins
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 667-9558
Fax	(505) 665-4414
E-mail	s.perkins@lanl.gov

Name	Reid B. Porter
Position	Graduate Research Assistant
Organization	Los Alamos National Laboratory
Phone	(505) 665-7508
Fax	(505) 665-4414
E-mail	rporter@lanl.gov

Name	John J. Szymanski
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 665-9371
Fax	(505) 667-0854
E-mail	szymanski@lanl.gov

Name	James Theiler
Position	Technical Staff Member
Organization	Los Alamos National Laboratory
Phone	(505) 665-5682
Fax	(505) 665-4414
E-mail	jtheiler@lanl.gov

Name	A. Cody Young
Position	Graduate Research Assistant
Organization	Los Alamos National Laboratory
Phone	(505) 667-9575
Fax	(505) 665-4414
E-mail	cody@lanl.gov

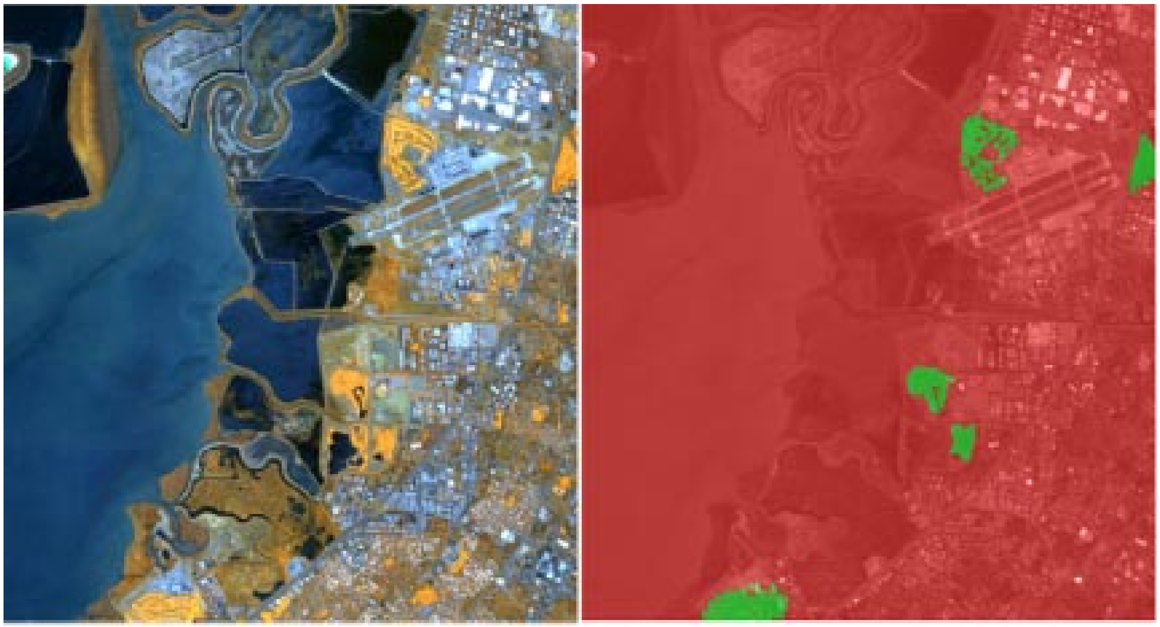


Figure 1. GENIE can find objects of interest in a range of remotely sensed imagery by searching for both spectral and spatial signatures of the object. An example is identifying golf courses at moderate resolution (~50 m × 50 m per pixel) in visible and near-infrared multispectral images, in which both the type of grass (a spectral feature) and the golf courses' characteristic shape (a spatial feature) define the object. Here GENIE cleanly extracts five golf courses (bright green areas) against a complex background of manmade and natural features, including parklands that are spectrally very similar to golf courses.

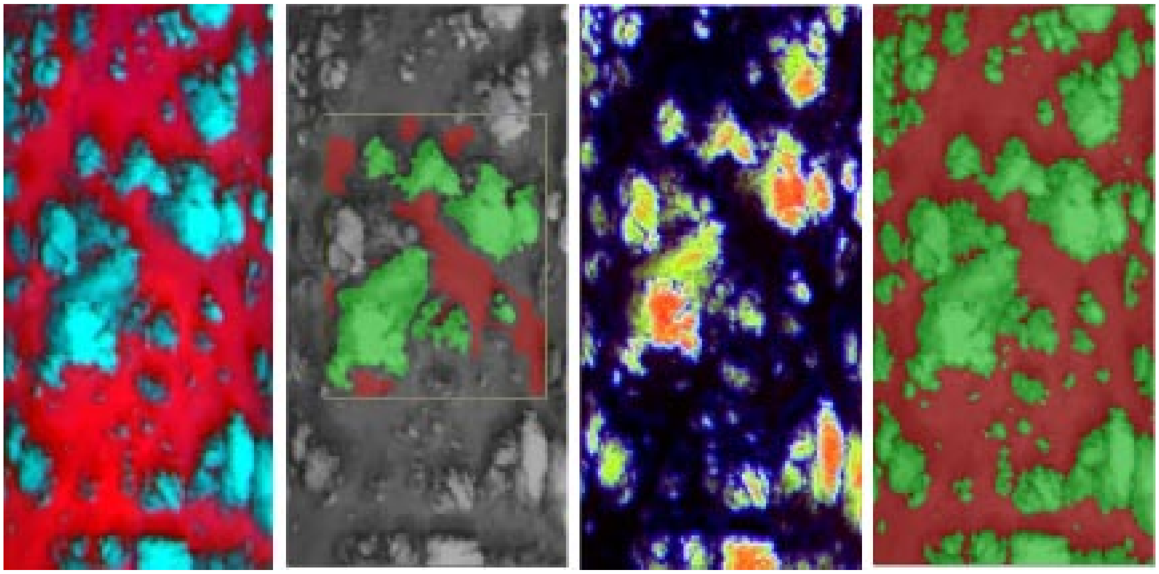


Figure 2. Automatic cloud detection is an important task for studies of the atmosphere in order to remove a common background feature in studies of Earth's surface. GENIE can identify clouds in thermal multispectral imagery (far left) on the basis of a small amount of training data (center left). In this example, GENIE evolved an algorithm that used spectral coolness/brightness and spatial edge detection to enhance clouds (center right) and then produce a cloud mask (far right) to remove them. The evolved algorithm showed great robustness in creating masks for other images. (From "Genetic refinement of cloud-masking algorithms for the multi-spectral thermal imager (MTI)," A. B. Davis, S. P. Brumby, N. R. Harvey, K. Lewis Hirsch, and C. A. Rohde, *Proceedings of IGARSS 2001*, Sydney, Australia, 9–13 July 2001.)

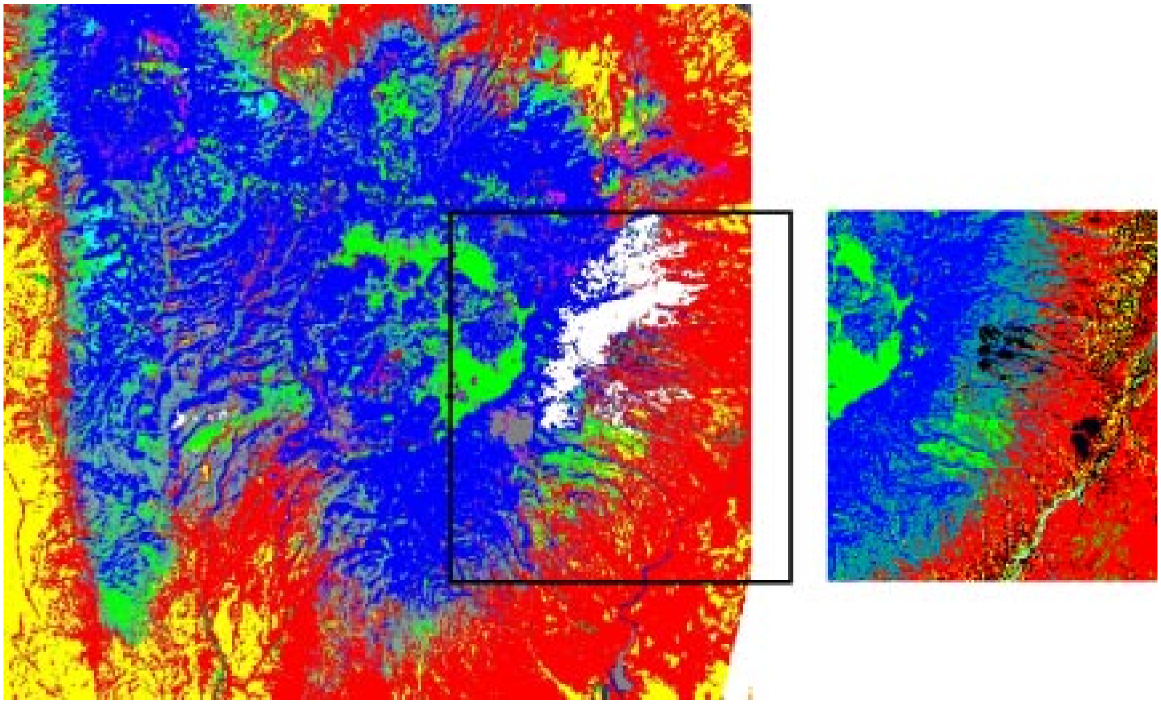


Figure 3. One of GENIE’s unique strengths is its ability to rapidly and repeatedly analyze large amounts of remotely sensed imagery. For example, the Cerro Grande wildfire significantly altered the face of the Jemez Mountains that overlook Los Alamos National Laboratory (LANL). The Lab’s prefire land-cover map (right), covering a 30-km by 30-km area, needed substantial revision on a short time scale (a few days) to enable planning for postfire restoration efforts. GENIE was able to learn from the old map and then update and extend the analysis of new images to cover the entire Jemez ecosystem (80-km by 60-km). The resulting map, which shows fire damage in white, is contributing to studies of long-term environmental changes (caused by wildfires, overgrazing, and logging) as well as to large-scale habitat studies for elk and threatened or endangered species unique to the Jemez Mountains. (From “Evolving land cover classification algorithms for multi-spectral and multi-temporal imagery,” Steven P. Brumby, James Theiler, Jeffrey J. Bloch, Neal R. Harvey, Simon Perkins, John J. Szymanski, and A. Cody Young. To appear in *Proceedings of SPIE* **4480**.)

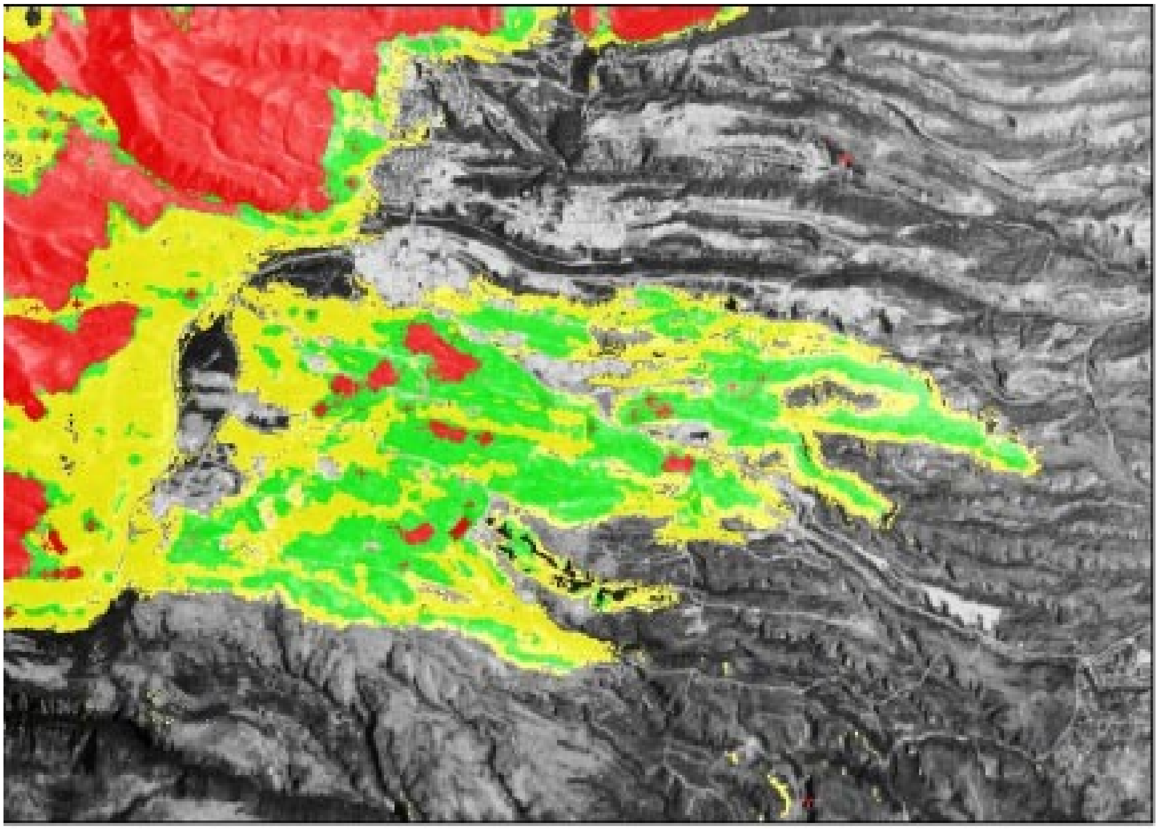


Figure 4. By running GENIE many times, the user can build up a multifeature classification map and set of extraction algorithms. In this example, GENIE was able to distinguish regions of high (red), medium (green), and low (yellow) burn severity from visible-light and thermal-infrared imagery gathered by an airborne sensor immediately after the Cerro Grande wildfire. These categories are standard classifications used by the United States Forest Service and correspond to regions of crown fire and high vegetation loss (high/medium severity) as opposed to regions of less-damaging underbrush fire (low severity). Identification of high-severity burn spots enabled prompt and accurate remediation work by ground crews and targeting for aerial reseeding flights. (From "Evolving forest fire burn severity classification algorithms for multi-spectral imagery," Steven P. Brumby, Neal R. Harvey, Jeffrey J. Bloch, James Theiler, Simon Perkins, A. Cody Young, and John J. Szymanski. To appear in *Proc. SPIE* **4381**. Presented at Aerosense 2001: SPIE's 15th Annual International Symposium on Aerospace/Defense Sensing and Controls, Orlando, Florida, April 16–20, 2001.)

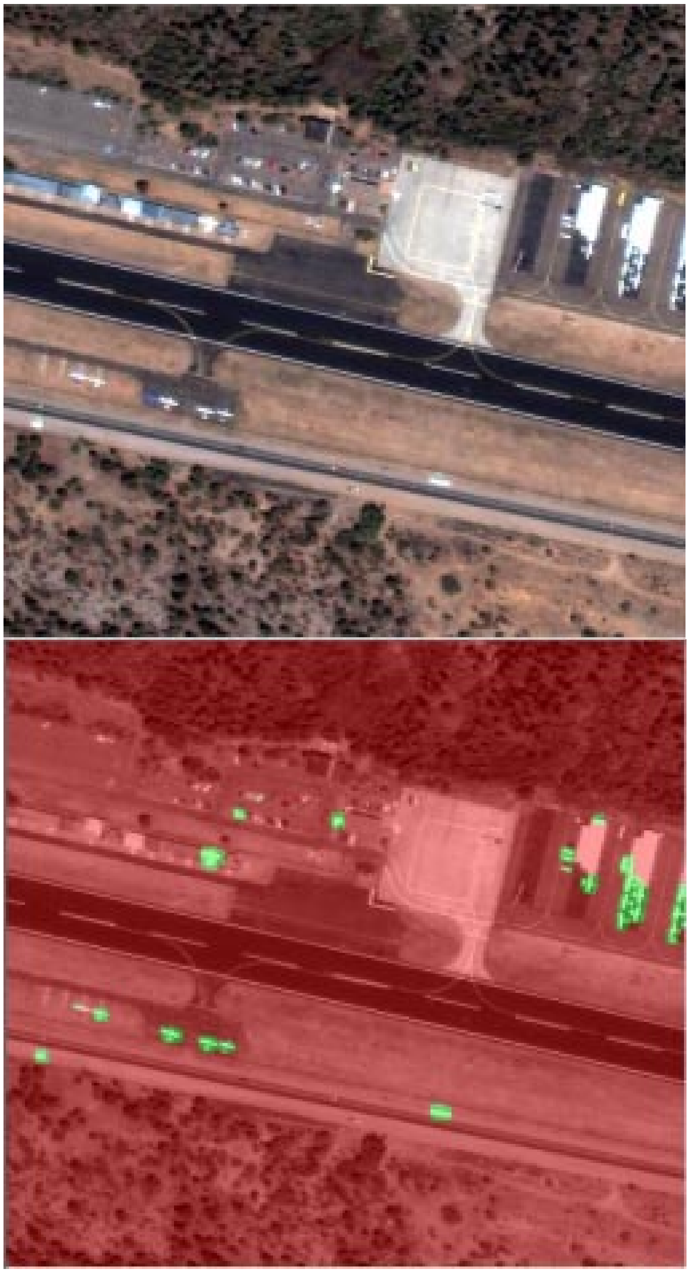


Figure 5. In addition to mapping large-scale land-cover features, such as forest or clouds, GENIE can be used to find compact, artificial features such as vehicles. This example compares a high-resolution color photograph of the Los Alamos County airport (top) and a mask created by GENIE (bottom). GENIE finds the small aircraft parked on the apron, as well as large trucks and brightly colored cars parked at the airport. This type of result can form the initial stage for higher-level analysis tools that identify objects by shape and context (e.g., adjacency) but which typically analyze only simple grayscale imagery, as opposed to the wide range of imagery handled by GENIE.

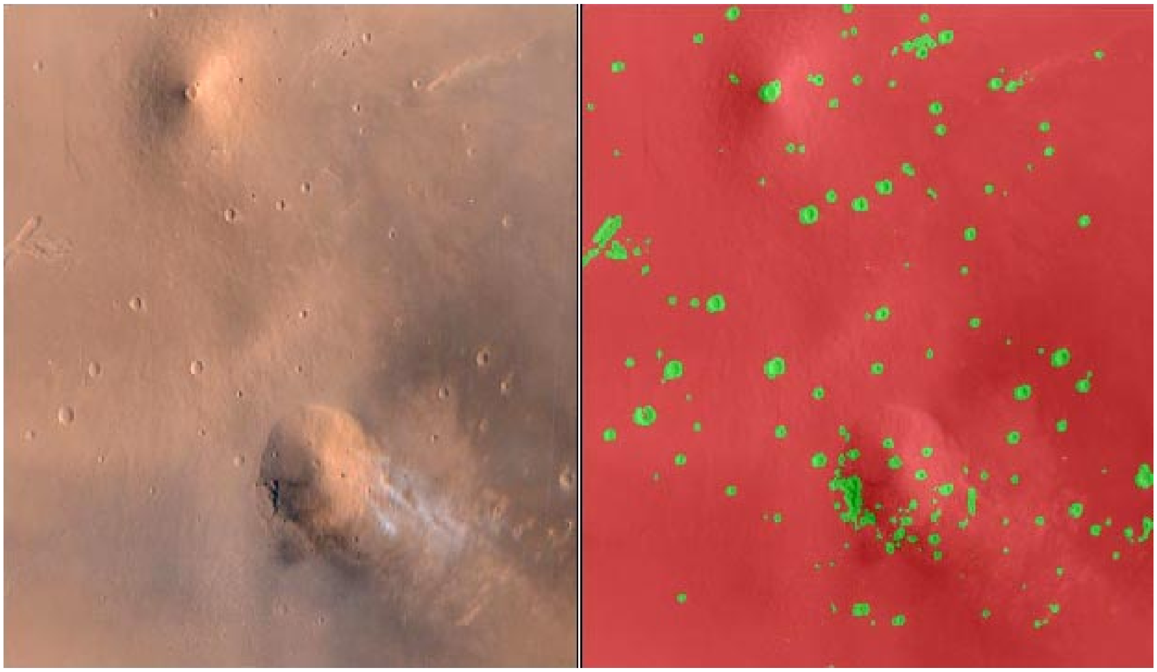


Figure 6. Analysis of remotely sensed imagery returned by planetary probes is one of the most exciting follow-on applications for the GENIE algorithms. For example, Mars is substantially smaller than Earth, but a total lack of surface water gives it the same land surface area as Earth's. The recent discovery of possible fluid erosion on the sides of some deep craters has implications for finding fossil or active life on Mars and has generated enormous interest in finding more examples of such features. Locating such fluid-erosion features will affect planning of future rover and sample-return missions. The first step to finding such features is to map craters (as shown). GENIE offers scientists studying Mars the opportunity to rapidly extract spatial features and land-cover types across Mars' entire surface, as well as track seasonal changes at the poles and in the aftermath of dust storms. (From "Automatic Feature Extraction for Panchromatic Mars Global Surveyor Mars Orbiter Camera Imagery," Catherine S. Plesko, Steven P. Brumby, and Conway Leovy. To appear in *Proceedings of SPIE* **4480**.)

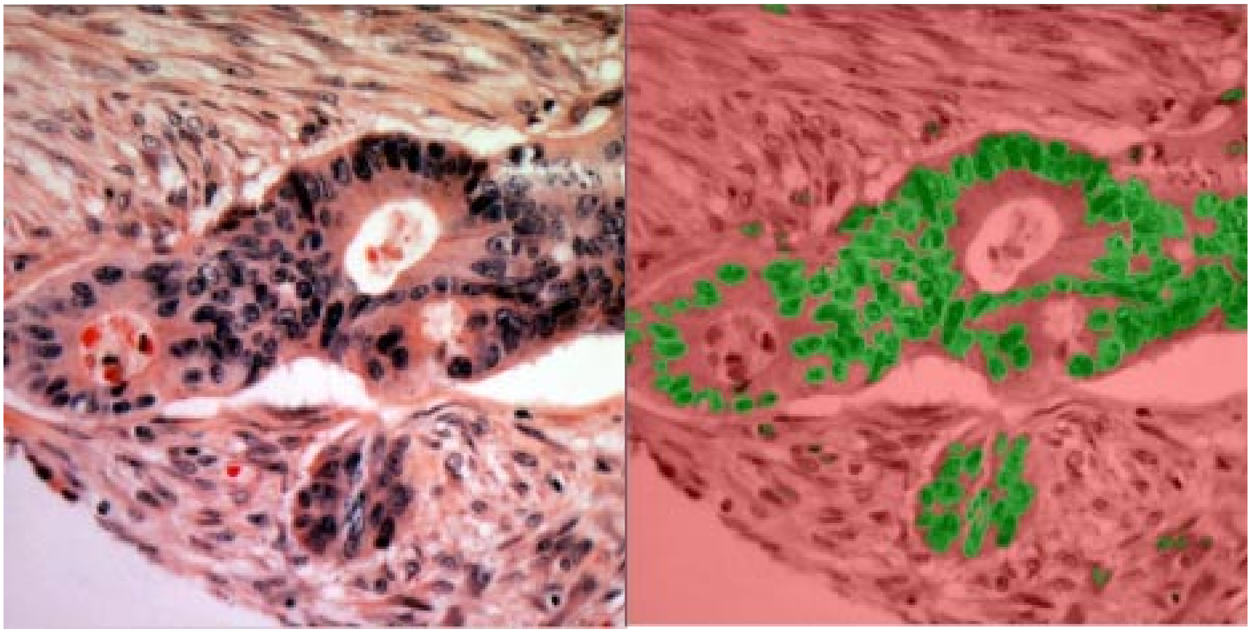


Figure 7. In addition to analyzing satellite images, GENIE can be applied to any other types of digital imagery. For example, in collaboration with CRI Inc. (http://www.cri-inc.com), we have begun using GENIE to detect cancer and other diseases in medical images. Shown here are a section of colon containing both cancerous and normal tissue (left photo), recorded by an advanced tunable multispectral imager attached to a microscope, and GENIE's identification of cancerous cells (green areas in right photo).

Comparison of GENIE and Conventional Supervised Classifiers for Multispectral Image Feature Extraction

Neal R. Harvey, James Theiler, Steven P. Brumby, Simon Perkins, John J. Szymanski,
Jeffrey J. Bloch, Reid B. Porter, Mark Galassi, A. Cody Young

Abstract— **We have developed an automated feature detection/classification system, called GENIE (GENetic Imagery Exploitation), which has been designed to generate image processing pipelines for a variety of feature detection/classification tasks. GENIE is a hybrid evolutionary algorithm that addresses the general problem of finding features of interest in multi-spectral remotely-sensed images. We describe our system in detail together with experiments involving comparisons of GENIE with several conventional supervised classification techniques, for a number of classification tasks using multi-spectral remotely-sensed imagery.**

Keywords— **Supervised Classification, Genetic Programming, Image Processing, Evolutionary Algorithms, Multispectral Imagery, Remote Sensing.**

I. Introduction

LARGE volumes of remotely-sensed multi-spectral data are being generated from an increasing number of increasingly sophisticated airborne and spaceborne sensor systems. While there is no substitute for a trained analyst, exploitation of this data on a large scale requires the automated extraction of specific features of interest. Creation and development of task-specific feature-detection algorithms is important, yet can be extremely expensive, often requiring a significant investment of time and effort by highly skilled personnel.

Our particular interest is the pixel-by-pixel classification of multi-spectral remotely-sensed images, not only to locate and identify but also to delineate particular features of interest. These range from broad-area features such as forest and open water to man-made features such as buildings and roads. The large number of features in which we are interested, together with the variety of instruments with which we work, make the hand-coding of suitable feature-detection algorithms impractical. We are therefore using a supervised learning approach that can, using only a few hand-classified training images, generate image processing pipelines that are capable of distinguishing features of interest from the background. We remark that our approach is to consider the two-class problem: although many applications require the segmentation of an image into a larger number of distinct land-cover types, we consider the simpler problem of identifying a single class against a background of "other" classes.

In applying general-purpose supervised learning techniques to multi-spectral imagery, the usual approach is to employ purely spectral input vectors, formed by the set of intensity values in each spectral channel for each pixel in the image. These vectors provide a convenient fixed-dimensionality space in which conventional classifiers can often work well. It is clear, however, that spatial relationships (such as texture, proximity, or shape, all of which are disregarded with purely spectral vectors) can be very informative in scene classification. Many different kinds of extra spatial context information could be added to the spectral information, as additional dimensions of the pixel input vector. The problem is that there exists a combinatorically vast choice for these additional vector dimensions; yet it is clear that a suitable choice of additional dimensions could make classification much easier. Unfortunately, this suitable choice is, in general, application-specific.

To address this problem, we have developed a hybrid evolutionary algorithm called GENIE (GENetic Imagery Exploitation) [2], [3], [4], [5], [6], [7], [8], that searches through the space of image processing algorithms. GENIE is a hybrid in that the evolutionary part of the program attempts to identify a pipeline of image processing operations which transform the raw multi-spectral data planes into a new set of image planes; these intermediate "scratch" planes are then used as input to a conventional supervised classification technique to provide the final classification results.

When adopting an evolutionary approach, a critical issue is the representation of candidate solutions in order that they may be effectively manipulated. We use a genetic programming (GP) method of representation of solutions, due to the fact that each individual will represent a possible image processing algorithm. GP has previously been applied to image-processing problems, including: edge detection [9], film restoration [10], face recognition [11] and image segmentation [12]. The work of Daida et al. [13] and Bandyopadhyay and Pal [14] (as well as our own work, cited above) is of particular relevance since it demonstrates that GP can be employed to successfully evolve algorithms for real tasks in remote-sensing applications.

The beauty of an evolutionary approach is its flexibility: all that is required is a representation for candidate solutions, a fitness measure for comparing candidate solutions, and a scheme for "mutating" candidate solutions into other candidate solutions. Many varied problems beyond image processing have been successfully solved using evolutionary computation, from optimizing of dynamic routing

This research was funded by the U.S. Departments of Energy and Defense

in telecommunications networks [15] to designing protein sequences with desired structures [16], and many others.

This paper describes our system in detail together with experiments involving comparisons of Genie with several conventional supervised classification techniques, for a number of classification tasks using multi-spectral remotely-sensed imagery.

The remainder of the paper is organized as follows: Section II describes the Genie system in detail. Section III describes the conventional supervised classification techniques with which Genie is to be compared. Section IV describes the data and classification tasks on which the algorithms are to be tested and compared. Section V describes the results of the comparisons. Section VI describes further comparison with multi-class versions of the supervised classifiers. Finally, section VII discusses these results and concludes.

II. The Genie System

Genie employs a classic evolutionary paradigm: a population is maintained of candidate solutions (*chromosomes*), each composed of interchangeable parts (*genes*), and each assessed and assigned a scalar fitness value, based on how well it performs the desired task. After fitness determination, the evolutionary operators of selection, crossover and mutation are applied to the population and the entire process of fitness evaluation, selection, crossover and mutation is iterated until some stopping condition is satisfied.

A. Training Data

The environment for each individual in the population consists of *data* planes, each of these planes corresponding to a separate spectral channel in the original image, together with a *weight* plane and a *truth* plane. The weight plane identifies those pixels to be used in training – these are the pixels for which the analyst is confident in identifying as either "true" and "false": *true* defines areas where the feature of interest exists; *false* defines areas where that feature does not exist. The actual delineation of true and false pixels is given by the truth plane. This arrangement permits us the flexibility (not used in this study) to employ both real-valued weights (representing degrees of confidence or of importance) and real-valued truth (corresponding to retrieval of continuous valued properties). The data in the weight and truth planes may be derived from actual ground truth (collected on the ground, at or near the time the image was taken) or from the best judgement of an analyst looking at the data. Because collecting ground truth data is so expensive, our system employs a graphical interface called Aladdin to assist the analyst in making judgements about and marking out features in the data. The analyst can view a multi-spectral image in a variety of ways, and can create training data by painting directly on the image using a computer mouse. Fig. 1 shows an image alongside the markup that an analyst provides as "ground truth." Figs. 4(b), and 6(b) show further examples where the analyst has marked out the desired feature on the image.

B. Encoding Candidate Solutions

Each individual *chromosome* in the population consists of a fixed-length string of *genes*. Each gene in Genie corresponds to a primitive image processing operation. Therefore the entire chromosome describes an algorithm consisting of a sequence of primitive image processing operations.

A single gene consists of an operator name, a list of input planes, specifying from which plane input is to come; a list of (usually one) output plane; and a list of scalar parameters. Parameters may be integer, floating point, or categorical. Each gene used in Genie takes one or more distinct image planes as input, and produces one or more image planes as output. Input can be taken from any data planes in the training data image cube. Output is written to any of a small number of *scratch planes* — temporary workspaces where an image plane can be stored. Genes can also take input from scratch planes, but only if that scratch plane has been written to by another gene earlier in the chromosome sequence.

The image processing algorithm represented by any particular chromosome can be thought of as a directed acyclic graph, where the non-terminal nodes are primitive image processing operations, and the terminal nodes are individual image planes extracted from the multi-spectral image used as input. The scratch planes are the "glue" that combines primitive operations into image processing pipelines. Traditional GP [17] uses a variable sized (within limits) tree representation for algorithms. Our representation differs in that it allows for re-use of values computed by sub-trees, since many nodes can access the same scratch plane, i.e., the resulting algorithm is a graph rather than a tree. It also differs in that the total number of nodes is fixed.

Our notation for genes is most easily illustrated by an example: the gene `[ADDP rD1 rS1 wS2]` applies pixel-by-pixel addition to two input planes, read from data plane 1 and from scratch plane 1, and writes its output to scratch plane 2. Additional operator parameters, if any, are listed after the input and output arguments.

Our "gene pool" is composed of a set of primitive image processing operators which we consider useful. For different applications, the user may want to choose different sets of primitive operators; for the studies described here, we used the operators described in Table I. These include spectral, spatial, spatio-spectral, logical and thresholding operators.

The set of morphological operators is restricted to function-set processing morphological operators, i.e. gray-scale morphological operators having a flat structuring element. The shape of the structuring elements used by these operators is chosen from among: square, circle, diamond, horizontal cross and diagonal cross, and horizontal, diagonal and vertical lines. The shape and size of the structuring element are defined by operator parameters. Other local neighborhood/windowing operators such as mean, median, etc. specify their kernels/windows in a similar way. The spectral operators have been chosen to permit weighted sums, differences and ratios of data and/or scratch planes.

It should be noted that although all chromosomes have the same fixed number of genes, the *effective length* of the

Gene Abbreviation	Image Processing Operation	Inputs/ Outputs/ Parameters	Notes
ADDP	Add planes	2/1/0	Basic mathematical operations. ADDS adds a scalar, which may be negative, to its input. DIFF is like SUBP but outputs the absolute values. NDI is like SUBP, but divides the result by the sum of its two inputs. MULTS scales its input by a scalar, which by default is positive. LINSCL is like MULTS but takes an extra parameter which is added onto the scaled input. LINCOMB outputs a linear combination of its two inputs, in proportion specified by its one parameter, which takes a value between 0 and 1.
ADDS	Add scalar	1/1/1	
SUBP	Subtract planes	2/1/0	
DIFF	Absolute difference	2/1/0	
NDI	Normalized difference	2/1/0	
MULTS	Multiply by scalar	1/1/1	
NEG	Negate plane	1/1/0	
MULTP	Multiply planes	2/1/0	
SQRT	Square root	1/1/0	
SQR	Square	1/1/0	
LINSCL	Linear scale	1/1/2	
LINCOMB	Linear combination	2/1/1	
MIN	Minimum	2/1/0	Logical operations. MIN and MAX perform pixel-wise minimum and maximum, equivalent to AND and OR for binary input. IFLTE outputs its third input wherever the first input is less than its second input, and its fourth input elsewhere.
MAX	Maximum	2/1/0	
IFLTE	'If less than else'	4/1/0	
CLIP_HI	Clip high	1/1/1	Thresholding operations. CLIP_HI truncates any pixel values above a value set by its parameter. CLIP_LO does the converse. THRESH sets all values below its threshold parameter to 0, and all those above to `dataScale`.
CLIP_LO	Clip low	1/1/1	
THRESH	Threshold	1/1/1	
SAVAR	Spectral angle variance	2-16/1/2	Spectral angle operations. SAVAR and SADIST look at two circular neighborhood regions around each pixel, of size defined by their two parameters. SAVAR returns the difference between the variance of the spectral angles of the pixels in the two regions. SADIF returns the difference between the mean spectral angle of both regions. SADIST returns the spectral angle difference between each pixel and the vector defined by its parameters. SANORM normalizes the vector defined by its inputs to have a magnitude equal to `dataScale`.
SADIF	Spectral angle difference	2-16/1/2	
SADIST	Spectral angle distance	2-10/1/2-10	
SANORM	Normalize spectral vector	2-10/2-10/0	
QTREG	Region Size related to Statistics	1/3/1	QTREG Determines the region size (in log base 2) around each pixel for which the normalized variance per pixel standard of the square region first reaches a given threshold. Also returns planes with the linear fit slope and offset of the variance as a function of region scale for each pixel
R5R5	Laws' texture measure	1/1/0	Neighborhood operations. In general, all these operations take a single plane as input and produce a single output plane. The output at each pixel is determined by looking at the pixel's neighborhood. R5R5, LAWB, LAWD, LAWF and LAWH are widely-used texture measures, developed by Laws, that return zero if the neighborhood contains all the same value of pixel, and some other value otherwise, depending upon the distribution of pixel values. R5R5 is corresponds to Laws' $R5^T \times R5$ 5×5 operator. The others are 3×3 operators, corresponding to Laws' $S3^T \times L3$, $E3^T \times E3$, $L3^T \times S3$ and $S3^T \times S3$ operators respectively. For details regarding Laws' textural operators, the interested reader is referred to [18], [19]. Most of the other operators are familiar image processing or morphological operators, whose description can be found in any good book on image processing. Most take two parameters which give the size and shape of a structuring element defining the neighborhood to which the operator is applied. ASF stands for 'Alternating Sequential Filter'. MB_EDGE takes an additional parameter defining a threshold for edge strength to be looked for. The single parameter for H_DOME and H_BASIN defines the pixel value offset used by these operators.
LAWB	Laws' texture measure	1/1/0	
LAWD	Laws' texture measure	1/1/0	
LAWF	Laws' texture measure	1/1/0	
LAWH	Laws' texture measure	1/1/0	
LAPLAC3	3x3 Laplacian	1/1/0	
LAPLAC5	5x5 Laplacian	1/1/0	
MORPH_LAPLAC	Morph. Laplacian	1/1/2	
ISO_GRAD	Isotropic gradient	1/1/0	
MEAN	Mean	1/1/1	
VAR	Variance	1/1/2	
SKEWNESS	Skewness	1/1/2	
KURTOSIS	Kurtosis	1/1/2	
SKEW_COEFF	Skewness coefficient	1/1/2	
KURT_COEFF	Kurtosis coefficient	1/1/2	
SD	Standard deviation	1/1/2	
RANGE	Morphological Gradient	1/1/2	
MEDIAN	Median	1/1/2	
EROD	Erode	1/1/2	
DIL	Dilate	1/1/2	
OPEN	Open	1/1/2	
CLOS	Close	1/1/2	
OPCL	Open-close	1/1/2	
CLOP	Close-open	1/1/2	
ASF_CLOP	ASF Close-open	1/1/2	
ASF_OPCL	ASF Open-close	1/1/2	
POS_TH	Positive top hat	1/1/2	
NEG_TH	Negative top hat	1/1/2	
OP_REC	Open with reconstruction	1/1/2	
CL_REC	Close with reconstruction	1/1/2	
H_DOME	H-dome	1/1/1	
H_BASIN	H-basin	1/1/1	
MB_EDGE	Canny edge detector	1/1/2	

TABLE I

The primitive image processing operators (genes) used in Genie and what they do.

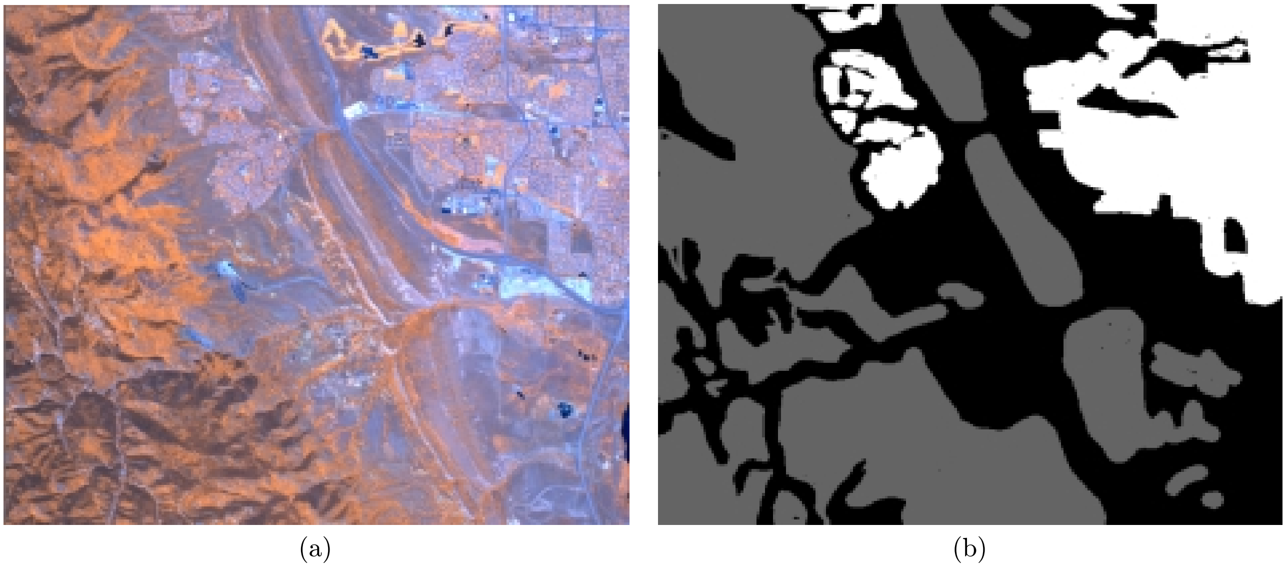


Fig. 1. (a) Gray-scale images of one of the scenes used to produce the training data for "Urban Areas" (Urban 1) (b) Training data provided for the training scene for "Urban Areas" (White = Feature, Grey = Not Feature, Black = No Assertion)

resulting algorithm graph may be smaller than this. For example, an operator may write to a scratch plane that is then overwritten by another gene before anything has a chance to read from it. Genie performs an analysis of chromosome graphs when they are created and only carries out those processing steps that actually affect the final result. Therefore, the fixed length of the chromosome acts as a maximum effective length.

In an interesting parallel to "junk DNA" in natural chromosomes, the final chromosomes produced by Genie often exhibit some redundancy, i.e. genes and answer planes that do not contribute to the answer. While these "junk genes" do not affect the functionality of the chromosome, they can make it harder to understand how the chromosome works. We have therefore developed a simple post-run pruning process that removes junk genes and ineffective answer planes from the final solution if this is required.

C. Backends

Final classification requires that the algorithm produce a single scalar output plane, which can then be thresholded to produce a binary output. It would be possible to treat, for example, the contents of scratch plane **S1** as the output from the algorithm (thresholding of this plane may be required to obtain a binary result). However, we have found it advantageous to adopt a hybrid approach which applies a conventional supervised classifier to a (sub)set of scratch and data planes to produce the final output plane.

To do this, we first select a subset of the scratch and data planes to be *answer planes*. The conventional supervised classifier "backend" uses the answer planes as input and produces a final output plane; in principle, we can use any supervised classification technique as the backend but for the comparisons reported here, we used the *Fisher Linear Discriminant* [20]. This provides a linear combination of the answer planes that maximizes the mean separation between true and false pixels, normalized by the total variance in the projection defined by the linear combination. The output of the discriminant-finding phase is a continuous-valued (gray-scale) image, which is then reduced to a binary image by finding the threshold value that maximizes the fitness as described in the following section.

D. Fitness Evaluation

The fitness of a candidate solution is given by the degree of agreement between the final binary output plane and the training data. If we denote the detection rate (fraction of "true" pixels classified correctly) as R_d and the false alarm rate (fraction of "false" pixels classified incorrectly) as R_f, then the fitness F of a candidate solution is given by

$$F = 500(R_d + (1 - R_f)). \quad (1)$$

Thus, a fitness of 1000 indicates a perfect classification result. This fitness score gives equal weighting to type I (true pixel incorrectly labelled as false) and type II (false pixel incorrectly labelled as true) errors. Note a fitness score of 500 can be trivially achieved with a classifier that identifies all pixels as true (or all pixels as false).

E. Software Implementation

The evolutionary algorithm code has been implemented in object-oriented Perl. This provides a convenient environment for the string manipulations required by the evolutionary operations and simple access to the underlying operating system (Linux). Chromosome fitness evaluation is the computationally intensive part of the evolutionary process and we currently farm this job out to a separate process running a commercial image processing engine (Interactive Data Language (IDL), by Research Systems, Inc. [21]). IDL does not provide all the image processing operators we want, so we have implemented additional operators in C that can be called from within the IDL environment.

Within IDL, individual genes correspond to single primitive image operators, which are coded as IDL procedures; a chromosome is a sequence of genes and exists as lines of IDL code in an IDL batch executable. In our present implementation, an IDL session is opened at the start of a run and communicates with the Perl code via a two-way UNIX pipe. This pipe is a low-bandwidth connection. It is only the IDL session that needs to access the input and training data (possibly hundreds of Megabytes), requiring a high-bandwidth connection. The Aladdin training data mark-up tool was written in Java. Fig. 2 shows the software architecture of the system.

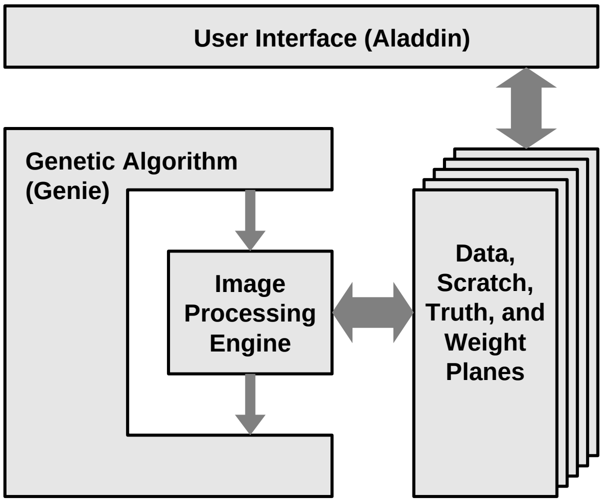

User Interface (Aladdin)
Genetic Algorithm (Genie)
Image Processing Engine
Data, Scratch, Truth, and Weight Planes

Fig. 2. Software Architecture of the Genie System.

III. Conventional Supervised Classification

Many implementations of standard supervised classifiers exist. One of the most widely used remote-sensing software packages is the ENvironment for Visualizing Imagery (ENVI) [1], which is built on IDL and is also distributed by Research Systems, Inc. Supervised classification techniques provided as part of the ENVI package were used in the comparison experiments with Genie. Currently Genie is set up to be trained using effectively three classes: "feature", "non-feature" and "don't care" and to be able to classify every pixel in its input data into one of two classes: "feature" and "non-feature". The normal mode of operation of the ENVI supervised classifiers is to use training data for the one "true" class: i.e. the feature of interest. The ENVI classifier is then used to classify the input image into "feature" or "unclassified". The user adjusts the parameters of the particular supervised classifier in order to attain optimal performance, with respect to feature identification. For our experiments, these parameters were adjusted to maximize the fitness defined in Eq. 1.

The one exception to this is the Maximum Likelihood classifier, which requires more than one class in the training data. In this case we used the "feature" and "non-feature" classes and the Maximum Likelihood classifier classified every pixel in the input data into one or other of these two classes, with no "unclassified" pixels being allowed. For applying the ENVI-supplied classifiers to out-of-training-sample data, the training data (reference spectra) used in the training was provided, together with the parameters that gave optimal performance on the training data. For the Genie case, it was simply a case of applying the algorithms found by Genie to the out-of-training-sample data (including the linear discriminant and threshold found during training).

In Section VI, we show auxiliary results from training the ENVI classifiers with more than just these two ("feature" and "non-feature") classes.

The following ENVI-supplied supervised classification techniques were used in the comparison experiments [22].

A. (MIN) Minimum Distance

The minimum distance supervised classification technique [22], [23] computes the mean pixel vector of the "feature" class, and then assigns new pixels to the "feature" class based on the Euclidean distance from that pixel to the mean. For the multi-class case, the pixel is assigned to the feature whose mean value is the minimum distance from the pixel. For the simple feature/non-feature discrimination here, the pixels is identified as a "feature" if the distance is less than a user-defined threshold (adjusted to obtain optimum performance on the training data); otherwise, it is a "non-feature".

B. (MAX) Maximum Likelihood

Maximum likelihood classification is the most common supervised classification method used with remote sensing data [23], and among the classifiers considered here, the one with the most free parameters. Here each class ("feature" and "non-feature") is modelled with separate multivariate gaussian distributions. New pixels are assigned to the class that had the highest probability of generating that pixel.

C. (MAH) Mahalanobis Distance

The Mahalanobis distance technique [23] is very similar to the maximum likelihood classifier, but with the simplification that all classes are modelled as having identical covariance matrices (which define the shape and orientation of the normal distribution). In the one class case, we compare the probability that a new pixel was generated by the "feature" class, to a user-defined threshold, in order to decide the class to which each pixel belongs.

D. (SAM) Spectral Angle Mapper

The spectral angle mapper (SAM) technique [24] is motivated by the observation that changes in illumination caused by shadows, slope variation, sun position, light cloud, etc., approximately only alter the magnitude of a pixel's vector, rather than the direction. Therefore we can eliminate these effects by normalizing all pixel vectors to unit magnitude and then looking at the angle between a given pixel and the mean vector for the "feature" class. Pixels are assigned to the "feature" class if this angle is less than a user-defined threshold.

E. (BIN) Binary Encoding

Binary encoding classification [23], [25] encodes the data and reference spectra into ones and zeros, based on whether a particular band value lies above or below the spectrum mean. The comparison between the encoded reference spectrum with the encoded data spectra is performed using a Boolean logic exclusive OR (XOR) function. A user specifies the minimum fraction of bands that must match between the encoded reference spectrum and the data spectra. Pixels that do not meet this criterion are labeled as "non-feature". We note that binary encoding produces an extreme coarsening of the data. It was invented for, and is most appropriately applied to, hyperspectral data.

It is worth noting that for the traditional supervised classifiers, the user-defined thresholds determined as being optimal for the training data may not be optimal for out-of-training-sample data. However, we can envisage a production scenario, where the classifiers are trained on one set of data to find a particular feature, where some kind of "ground truth" is available and the resultant classifier is applied to some other out-of-training-sample data, in order to determine if that particular feature is present or not in the data, and "ground truth" data not be available for that data. In this case, the lack of ground truth means that there is no quantitative way of determining the optimal threshold value for the out-of-training-sample data. It should also be pointed out that this is also the case for the Genie classifiers. Genie's backend has a threshold which needs to be determined and the value determined as optimal for a training set may not be optimal for out-of-training-sample data. So, for a fair comparison, thresholds determined for all classifiers during training where left unchanged when the classifiers were applied to out-of-training-sample data. In addition, experiments were also conducted in which user-adjusted thresholds were not employed, where the traditional classifiers were forced to classify the entire scene into feature or non-feature based on the particular distance measure appropriate to the classifier. This amounts to a planar separating surface compared to a sphere for the user-defined threshold case. It was found that the user-adjusted threshold scenario performed better, in general.

IV. Experimental Data and Classification Tasks

A. Data Used in the Experiments

The remotely-sensed images referred to in this paper were derived from the Airborne Visible and InfraRed Imaging Spectrometer (AVIRIS) [26], a sensor developed and operated by the NASA Jet Propulsion Laboratory. The AVIRIS sensor collects data in 224 contiguous, relatively narrow (10 nm), uniformly-spaced spectral channels. AVIRIS is an airborne sensor and spatial resolution can vary from a few meters to 20 meters, depending on the altitude of the collecting platform. We used data from 1996 and 1997 AVIRIS campaigns from a range of sites shown in Table II; more detail is available from the AVIRIS quick-look website [27].

For the studies reported here, we used a reduced number of relatively wide spectral bands, designed to simulate imagery from a new remote sensing satellite called the Multispectral Thermal Imager (MTI) [28]. The MTI satellite was launched in March 2000 and collects data in 15 spectral bands. Ten of these bands sample wavelengths between 0.4 and 2.4 microns, a region covered by the AVIRIS instrument. As test data to develop analysis codes for the MTI mission, AVIRIS data were convolved with the MTI spectral filter functions to produce simulated MTI data. This 10-band simulated data was used for development of both conventional remote sensing algorithms and for Genie development, such as reported here.

Feature	Site Name	AVIRIS Flight Number
Roads 1	Moffet Field	f970620t01p02_r03_sc01
Roads 2	ARM Site	f970801t01p02_r01_sc06
Roads 3	Denver	f970701t01p02_r07_sc07
Golf 1	Moffet Field	f970620t01p02_r03_sc01
Golf 2	Denver	f970701t01p02_r07_sc01
Golf 3	Kennedy Space Center	f960323t01p02_r04_sc02
Urban 1	Denver	f970701t01p02_r07_sc09
Urban 2	Denver	f970701t01p02_r07_sc04
Urban 3	Moffet Field	f970620t01p02_r03_sc01
Clouds 1	Atlanta	f970806t01p02_r07_sc01
Clouds 2	Atlanta	f970806t01p02_r07_sc02
Clouds 3	"Clouds 1"	f970817t01p02_r05_sc01

TABLE II
List of Data Sets Used in the Experiments

The images displayed here are false-color images (which have then been converted to gray-scale in the printing process). The color mappings used are the same for all original image data shown. The particular color mappings used here involve averaging MTI bands A (0.45–0.52 μm) and B (0.52–0.60 μm) for the blue component, bands C (0.62–0.68 μm) and D (0.76–0.86 μm) for the green component and bands E (0.86–0.89 μm) and F (0.91–0.97 μm) for the red component. In addition, the images have been contrast enhanced. The choice of color mappings was arbitrary, in that it was a personal decision made by the analyst in order to best "highlight" the feature of interest, and thereby enable the production of high quality training data. This ability to manipulate the image with color mappings and contrast enhancement is an important feature of the graphical interface.

B. Classification Tasks

We chose four different features of interest: roads, golf courses, urban areas, and clouds. These features were chosen because of their particular attributes in multi-spectral data. The features were considered a good test of a supervised classification technique due to the different levels of difficulty they posed for these techniques. Clouds are relatively easy, and mostly spectral; urban areas encompass a land-cover distinction; roads are easy for the eye to find, but notoriously difficult for automated algorithms; golf courses require a combination of spectral and spatial information to disambiguate them from other similarly-vegetated areas (e.g. lawns).

We set the various supervised classification techniques the task of distinguishing these features within several scenes of the 10-channel multi-spectral data as described above. For each feature of interest three separate scenes had training data marked-up using the Aladdin tool. This provided "ground truth" for training data and for assessing the performance of the classification scheme on out-of-training-sample data. We employed a cross-validation scheme where, for each feature, we trained a classifier separately on the three marked-up scenes, and then for each scene, applied the resulting classifier to the two remaining out-of-sample scenes. Genie was run, with a population of 100 individuals, for 500 generations, or until a (perfect score) fitness of 1000 was achieved.

An example of an image plus associated training data is shown in Fig. 1. This figure shows the false-color image for one of the scenes used for the "urban area" feature classification, and the associated training data. In the training data image the white pixels correspond to the places on the image where the feature is asserted to be, the grey pixels to where the feature is asserted not to be, and the black pixels correspond to places where no assertion is made.

V. Comparison Experiments

For the training phase, we ran Genie and the ENVI-supplied classifiers on the training data. For GENIE, the result of this training phase is an image processing pipeline which can be applied to and tested on other data. To apply the ENVI-supplied classifiers to out-of-training-sample data it was necessary to save the regions of interest of the marked-up training classes and provide them as the reference spectra for application of the classifiers to out-of-training-sample data.

We measured the fitness, detection rate and false-alarm rate of all the classifiers on the training data and out-of-training-sample data. Table III summarizes the quantitative results of the comparison between the Genie algorithm output and the traditional algorithms' output for each of the features. The bottom four rows of the table show the average, for each classification technique, across all features sought. It is interesting to notice that the relative ranking (based on fitness score) of each of the classifiers is relatively stable over the different features, with the more complicated classifiers generally achieving the highest scores. For the out-of-training-sample data, by contrast, the simpler algorithms (with fewer free parameters) perform much better. The main exception is Genie, which performs well on both the traning data and on the out-of-training-sample data.

An example of an image processing pipeline produced by Genie is given by the following solution to the golf course-finding task:

```
[QTREG rD7 wS5 wS3 wS1 0.05]*
[MEAN rD4 wS2 4 0]
[MEAN rS2 wS2 3 0]
[VAR rD7 wS4 3 0]
[CLOP rS2 wS2 3 0]
[RANGE rD10 wS1 3 0]
[OP_REC rS4 wS4 3 0]
[ASF_OPCL rD2 wS3 3 0]*
```

As described in Section II-B, each line consists of a single primitive image processing operation: the name of the operator, which data (D) or scratch (S) planes were read (r) from and which were written (w) to, and what parameter values were used (see Table I for details on the individual operators). Genie produced a solution with five answer planes, and the backend produced a linear combination of those planes, along with a threshold value, to give a binary classification. A graphical representation of this pipeline is illustrated in Fig. 3. Note that the circled **D**s represent the input data planes and the circled **S**s represent the answer planes that are input to the back-end classifier (Fisher Linear Discriminant plus threshold), to produce the final classification result. To aid clarity, we now provide a narrative description of the operation of this pipeline.

The RANGE operator computes the difference between the maximum and minimum value in a 7×7 kernel of data plane D10, and writes the result to scratch plane S1. The parameters "3 0" correspond to a square 7 × 7 kernel. The first integer parameter for this operator, "3", actually defines the "radius" of the smoothing kernel, where the "diameter" of the kernel is always an odd integer, and defined as (2 × radius) + 1. The second integer parameter, "0", defines the particular choice of kernel shape, in this case a square. A "1" would define a circle, "2" a vertical cross, "3" a diagonal cross, etc..

The first MEAN operator, [MEAN rD4 wS2 4 0], smooths the data plane D4 with a 9 × 9 square kernel, and writes the solution to scratch plane S2. The second MEAN operator, [MEAN rS2 wS2 3 0], smooths the result stored in the S2 plane with a 7 × 7 square kernel, and the CLOP operator performs a morphological close-open operation, again with a 7 × 7 square kernel, writing the output to scratch plane S2.

The ASF_OPCL operator performs an alternating sequential open-closing with a square kernel of maximum size 7×7 on data planeD2, and writes the output to scratch plane S3.

The VAR operator computes the variance in a 7 × 7 kernel of data plane D7, and writes the result to scratch plane S4. That plane is further modified by the OP_REC operator, which performs a morphological opening with reconstruction, again based on a 7 × 7 kernel.

The QTREG operator also reads data plane D7 and writes three scratch planes (S5, S3, and S1), two of which (S3 and S1) are overwritten by other operators before being used.

Finally, the Fisher Discriminant backend applies a linear combination of the scratch planes, followed by a threshold, to produce a binary answer plane. The coefficients applied to the five answer planes (S0, S1,S2, S3, S4) are: $\{-9.354 \times 10^{-6}, 1.235 \times 10^{-5}, 1.659 \times 10^{-6}, 1.460 \times$

		Training Data						Out-of-Training-Sample Data					
		Genie	MIN	MAH	MAX	SAM	BIN	Genie	MIN	MAH	MAX	SAM	BIN
Roads	Fitness	963.3	781.2	865.9	921.1	803.9	677.5	763.2	559.9	500.0	611.2	566.0	587.1
	DR (%)	96.61	78.63	83.11	91.62	82.93	82.00	60.71	27.54	0.00	62.25	30.29	72.94
	FAR (%)	3.95	22.40	10.03	7.40	22.14	46.50	7.36	15.57	0.00	40.02	17.09	55.51
	Rank	1st	5th	3rd	2nd	4th	6th	1st	5th	6th	2nd	4th	3rd
Golf	Fitness	998.3	945.0	947.6	966.2	915.4	820.1	739.8	584.8	500.0	553.1	696.5	572.9
	DR (%)	99.68	95.16	93.84	96.21	92.73	78.72	61.60	42.34	0.00	10.93	58.76	51.16
	FAR (%)	0.00	6.17	4.32	2.97	9.64	14.71	13.65	25.39	0.00	0.31	19.46	36.57
	Rank	1st	4th	3rd	2nd	5th	6th	1st	3rd	6th	5th	2nd	4th
Urban	Fitness	998.9	694.7	861.6	963.6	636.3	580.4	813.5	586.2	514.6	569.4	499.1	521.9
	DR (%)	99.85	58.55	80.34	95.67	75.03	83.59	66.32	27.36	2.93	65.86	50.67	70.18
	FAR (%)	0.07	19.61	8.03	2.94	47.77	67.52	3.63	10.11	0.02	51.97	50.51	65.80
	Rank	1st	4th	3rd	2nd	5th	6th	1st	2nd	5th	3rd	6th	4th
Clouds	Fitness	999.9	975.7	946.7	997.9	979.4	760.2	978.0	968.6	632.0	701.7	975.3	727.1
	DR (%)	99.99	96.41	94.21	99.91	98.18	55.59	97.43	95.38	28.62	99.97	97.28	48.85
	FAR (%)	0.00	1.27	4.86	0.33	2.29	3.55	1.82	1.67	2.22	59.64	2.23	3.43
	Rank	1st	4th	5th	2nd	3rd	6th	1st	3rd	6th	5th	2nd	4th
Average	Fitness	990.1	849.1	905.5	962.2	833.8	709.5	823.6	674.9	536.6	608.8	684.2	602.3
	DR (%)	99.03	82.19	87.87	95.86	87.22	74.98	71.51	48.16	7.89	59.75	59.25	60.78
	FAR (%)	1.01	12.36	6.81	3.41	20.46	33.07	6.62	13.18	0.56	37.99	22.32	40.33
	Rank	1st	4th	3rd	2nd	5th	6th	1st	3rd	6th	4th	2nd	5th

TABLE III

Comparison of Genie's Evolved Algorithm with ENVI Algorithms (DR = Detection Rate, FAR = False Alarm Rate)

$10^{-9}, 0.0349\}$. There is an additional DC offset value of -0.350 applied to the output of the linear combination. The threshold value for determining the binary output was 0.664305.

It can be seen that this image processing pipeline has only used 4 of the available 10 data planes as input: data planes D2, D4, D7 and D10. These correspond to the MTI bands B (0.52–0.60 μm), D (0.76–0.86 μm), G (0.99–1.04 μm) and O (2.08–2.35 μm) respectively. Genie's choice of input data bands is (in retrospect) not too surprising, given the task. The algorithm is using the green band (B), as well as two near-infrared (NIR) bands (D,G) and a short-wave infra-red (SWIR) band (O). Vegetation is highlighted in the two NIR bands that Genie selected, as well as in the green band.

Of these five answer planes the most important were S1, S2 and S4; using only those planes we could still achieve the same fitness value, on the training data and out-of-training-sample data, as when all the answer planes were used. Hence, two of the operators (those marked above with asterisks) did not contribute substantially to the solution. The outputs of the useful answer planes, as can be seen from Fig. 3, are derived from the NIR and SWIR bands. In this case we see, somewhat surprisingly, that the green band is not contributing significantly to the solution. We might expect green to be very useful for identifying golf courses, and this is probably how it made its way into the chromosome. However, in the end, the NIR and SWIR bands were found to be more informative.

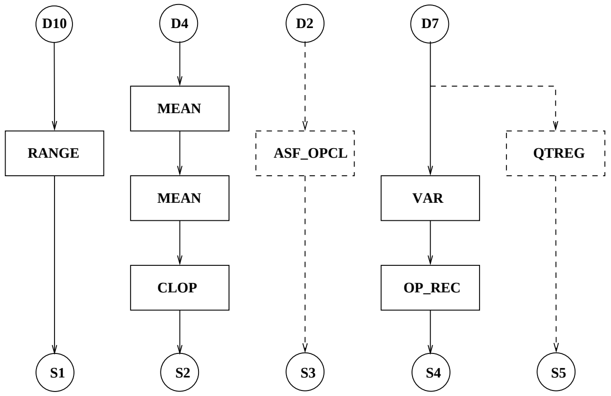


Fig. 3. Image processing pipeline discovered by Genie for finding golf courses. Dotted lines indicate scratch planes which did not contribute significantly to the final classification.

We illustrate the results of these classification techniques on training and out-of-training-sample data with an example of output from Genie, and from the best-performing ENVI classifier, on the golf course problem. Figs. 4 and 5 compare Genie to Maximum Likelihood (MAX) for one of the training data sets, and Figs. 6 and 7 compare Genie to the Spectral Angle Mapper (SAM) on out-of-training-sample data.

An interesting aspect of Genie's performance to consider is its repeatability: i.e. whether or not, for a given feature, Genie leads to the same result (i.e., the same "image processing pipeline") when trained on different scenes. In

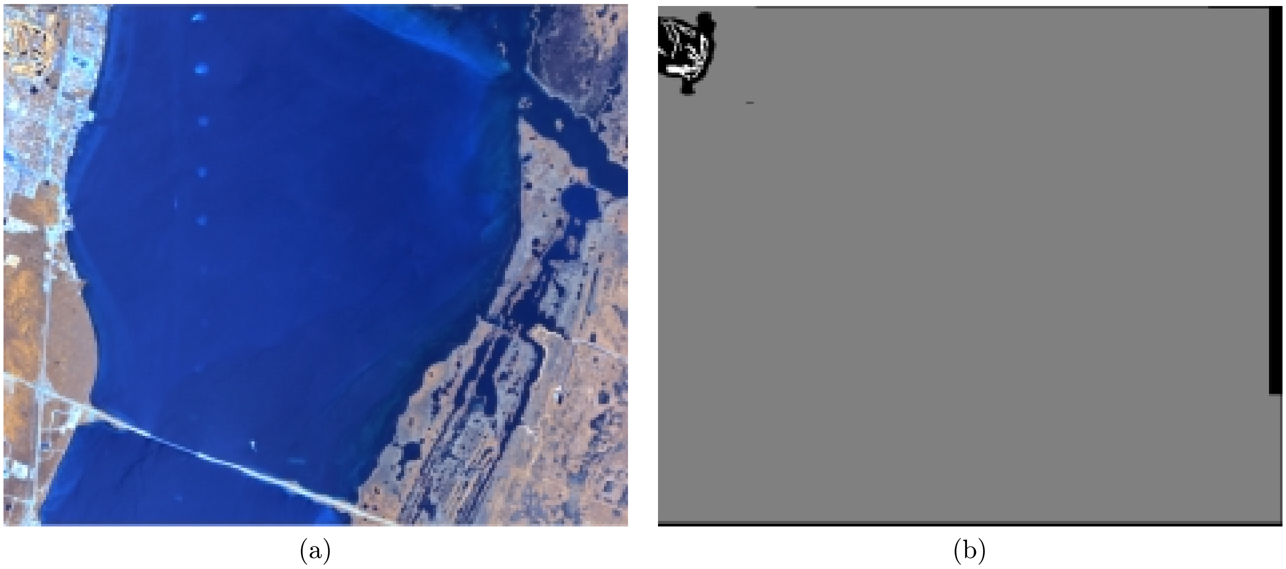

(a) (b)

Fig. 4. (a) Gray-scale images of one of the scenes used to produce the training data for "Golf Courses" (Golf 3) (b) Training data provided for the training scene for "Golf Courses" (White = Feature, Grey = Not Feature, Black = No Assertion). The black "buffer area" around the golf course reflects the analyst's lack of concern with a detailed delineation of the precise extent of the golf course.

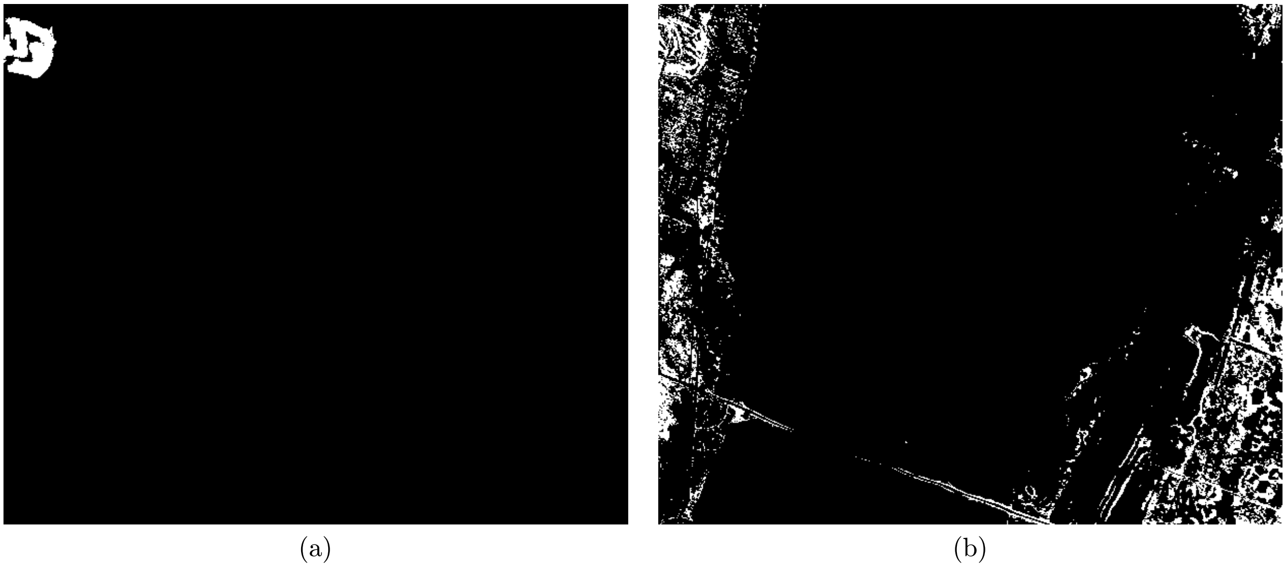

(a) (b)

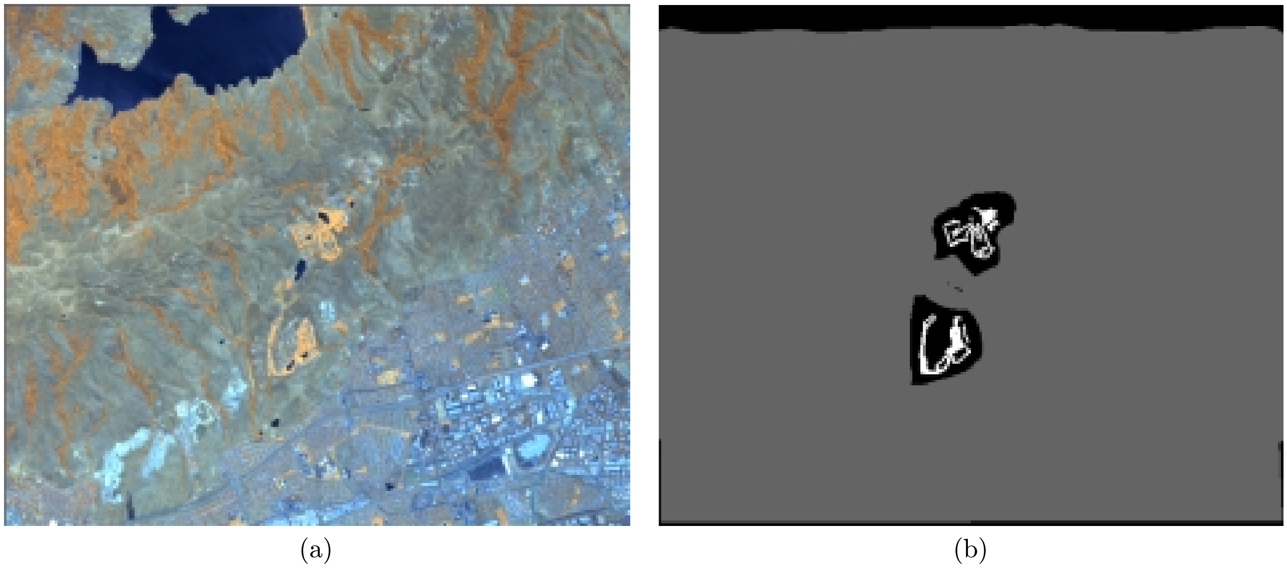
Fig. 5. (a) Genie results on training data: Fitness = 999.2 (b) Best ENVI classifier for the particular training scene (Minimum Distance): Fitness = 957.4. Here Genieś use of spatial information is clearly evident. The ENVI classifier actually did a better job of delineating the extent of the golf course, whereas Genieś spatial operators led to a "fatter" golf course than the purely spectral data would warrant. On the other hand, though, this spatial information allowed Genie to veto the golf course-like spectra in the rest of the image. Because the "fatter" golf course fits inside the no-assertion region, Genie is not penalyzed.

general, Genie will not produce the same image processing pipeline even when trained on the same scene, if it starts with a different random number seed. However, the different solutions will generally have the same approximate performance, both on training data and on out-of-sample data, and there will often be an overlap in the choice of operators and data planes used in the image processing pipeline that is evolved. But the space of image processing pipelines it too large and too rugged to achieve any real level of "robustness", in this regard.

VI. Further Experiments and Results

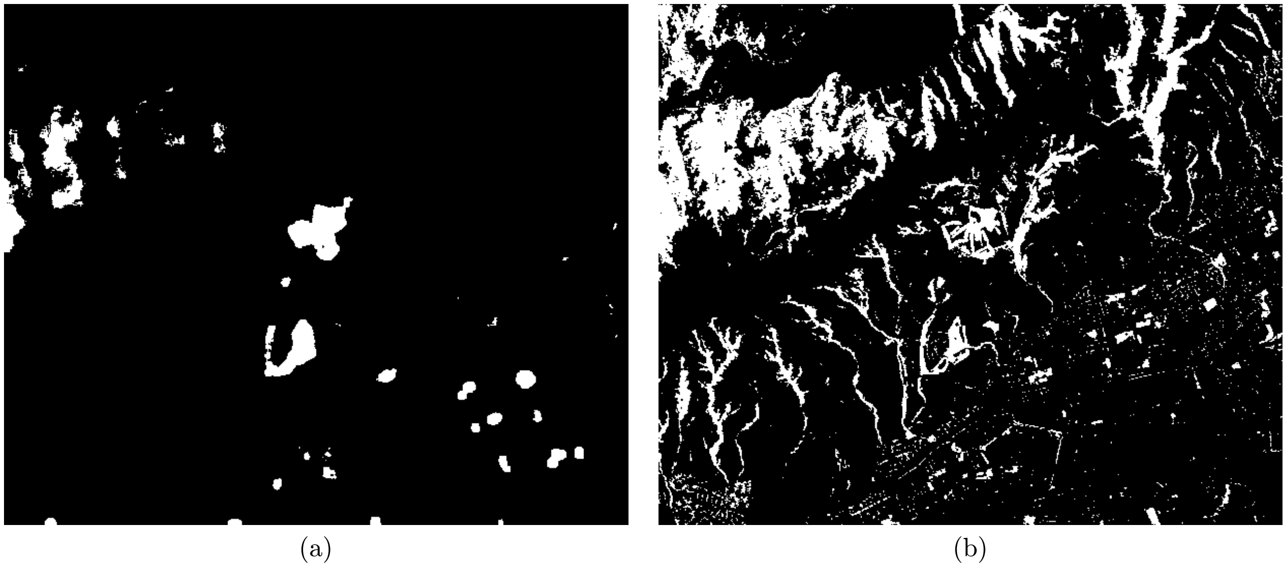
Depending on the application at hand, an image analyst is sometimes interested in the identification of a single specific feature against a background of everything else in the image, and is sometimes interested in the simultaneous extraction of multiple features (for instance, when making a landcover map). The experiments described in the previous sections take the first point of view and it is this binary classification task that Genie was designed to handle.

However, for Maximum Likelihood and other conventional classifiers, the "background of everything else" is not well modelled as a single unimodal class. To address



(a) (b)

Fig. 6. (a) Gray-scale images of one of the scenes used to produce training data for "Golf Courses" (Golf 1) (b) Training data provided for the training scene for "Golf Courses" (White = Feature, Grey = Not Feature, Black = No Assertion)



(a) (b)

Fig. 7. (a) Genie results on out-of-training-sample data: Fitness = 946.9 (b) Best ENVI classifier (for particular training scene) on out-of-training-sample data (Spectral Angle Mapper): Fitness = 856.7. Again, Genie has used its spatial operators to produce "fatter" golf courses, but it was also able to censor more of the non-golf-course area in the rest of the scene.

this difficulty, it has been suggested [29] to artificially divide the background into multiple classes, and then employ multi-class classification techniques. This combined use of labelled and unlabelled samples can often lead to more powerful supervised classification [30], [31], [32], [33].

In order to address these same issues, we conducted a series of further experiments where we adopted a similar approach, in which the standard supervised classification techniques were given the task of classifiying the scenes into multiple classes instead of the two feature/non-feature classes described in the previous experiments.

A. Experimental Procedure

The training data as provided to Genie and as used in the experiments described in Section V were used to create the training data provided to the standard supervised classifiers. The "feature" class was kept as it was, but the "non-feature" class was divided up into multiple classes. The combination of the "feature" class and the sub-divided "non-feature" class was then given as training data to the standard supervised classification techniques.

It should be noted that the binary encoding supervised classification technique was not included in these additional experiments.

The "non-feature" class was divided into multiple classes by applying ENVI's unsupervised k-means classification algorithms [1], [34], [35] to the entire "non-feature" class. This k-means classification was performed several times, varying the number of classes into which the non-feature class was classified. The k-means classification with the number of classes that provided the best final classification performance in terms of fitness, was the one included in the additional results shown here.

Table IV shows the overall ranking for the multiple-class classification algorithms, averaged over all the features for the training data. In this table, "(M)" indicates the use of multi-class training; the non-multiple-class results are those results described and shown in Section V.

	Training		Out-of-Training-Sample	
Rank	Classifier	Fitness	Classifier	Fitness
1st	Genie	990.1	Genie	824.5
2nd	MAX (M)	965.8	SAM (M)	780.0
3rd	MAH (M)	949.1	MIN (M)	778.7
4th	MAH	905.5	MAH (M)	752.4
5th	MIN (M)	892.3	SAM	684.2
6th	SAM (M)	874.1	MIN	674.9
7th	MIN	849.1	MAX (M)	615.5
8th	SAM	833.8	MAH (M)	536.6

TABLE IV

Rankings, based on fitness scores averaged over all classification tasks, of the best multiple-class variants of the standard supervised classifiers; on both training data and testing data

VII. Discussion

With a single exception, Genie out-performed all the other classification techniques on both training data and out-of-training-sample data, for all of the classification tasks considered. For the training data, the gap, with respect to fitness, between Genie's performance and the best of the other techniques was much less than for the out-of-training-sample case. This suggests that Genie is significantly better at generalizing than the other techniques compared here. An interesting observation is that the best of the other techniques on the training data did not necessarily guarantee it to be the best of the other techniques on the out-of-training-sample data. This indicates the sensitivity of these techniques to training data and highlights Genie's generalization abilities.

The one exception was the multi-class Spectral Angle Mapper applied to golf courses, on out-of-training-sample data. This suggests that golf courses are relatively well identified by their spectral signatures (perhaps not surprising in a desert/mountain environment where they are quite distinctive), and that the illumination-invariance built into the Spectral Angle Mapper provided it the edge to better generalize to other scenes. Since Genie was trained on only one scene at a time, it did not "learn" to employ an illumination-independent solution.

One issue to be addressed is training time. At present Genie requires the testing of potentially thousands of candidate algorithms on the training data. Depending on the size of the data, this can take hours to complete. This is considerably longer to train than the other techniques. It should be noted, though, that the result of Genie's training is an image processing algorithm that can be applied to other data with times comparable to that of the other techniques' application to out-of-training-sample data. We also remark that a few hours is usually a small fraction of the time it would take to hand-design an equivalent image processing pipeline that is customized not only to the specific feature, but also to the specific data set. Another point to consider is that being a population-based optimization technique, Genie lends itself well to parallelization, which can dramatically reduce training time. Some experiments have been carried out to demonstrate this [6].

Although the traditional classification techniques that were compared here use only spectral information, it is possible to enable these techniques to use spatial information as well. There is in fact a large literature on methodologies for combining spatial and spectral information (eg, see Refs. [36], [37], [38], [39]). Our approach was to apply a set of spatial operators to each plane in the input multispectral data and then combine these new processed data planes with the raw data planes; both sets of planes would then be provided as input to the supervised classifiers. We applied a number of morphological smoothings at different scales to the input data and combined this with the original data. We found that this information did improve the fitness scores achieved by the conventional supervised classifiers, but they were still considerably below the performance of Genie on the original data. Also, the improved performance was only for the training data. The classifiers actually performed worse on out-of-training-sample data (i.e., they were less robust). Obviously, if one were to adopt this approach, the choice of which spatial operators to apply is very important and the search space in this regard is immense. If one considers a scenario where some sophisticated technique is used to search the space for the optimal combination of spatial operators, one is entering the arena in which Genie is designed to function.

In conclusion, an automated feature detection/classification system based on genetic programming has been described. Experiments comparing this new system with traditional supervised classifiers indicate consistently better performance, on both training data and out-of-training-sample data. We attribute Genie's success to the choice of solution representation – as a multispectral image processing pipeline – and to the fact that it very naturally combines information from both the spectral and spatial domains.

References

[1] `http://www.rsinc.com/envi/index.cfm`

[2] S.P. Brumby, J. Theiler, S.J. Perkins, N.R. Harvey, J.J. Szymanski, J.J. Bloch and M. Mitchell, "Investigation of Image Feature Extraction by a Genetic Algorithm," *Proc. SPIE*, vol 3812, pp. 24–31, 1999.

[3] J. Theiler, N.R. Harvey, S.P. Brumby, J.J. Szymanski, S. Alferink, S. Perkins, R. Porter and J.J. Bloch, "Evolving Retrieval Algorithms with a Genetic Programming Scheme," *Proc. SPIE*, vol 3753, pp. 416–425, 1999.

[4] N.R. Harvey, S. Perkins, S.P. Brumby, J. Theiler, R.B. Porter, A.C. Young, A.K. Varghese, J.J. Szymanski and J.J. Bloch, "Finding Golf Courses: The Ultra High Tech Approach," *Lecture Notes in Computer Science*, vol. 1803, pp. 54–64, 2000.

[5] S.P. Brumby, J. Theiler, N.R. Harvey, S.J. Perkins, R.B. Porter, J.J. Szymanski and J.J. Bloch, "A Genetic Algorithm for Combining New and Existing Tools for Multispectral Imagery," *Proc. SPIE*, vol 4049, 2000.

[6] N.R. Harvey, S.P. Brumby, S.J. Perkins, R.B. Porter, J. Theiler, A.C. Young, J.J. Szymanski and J.J. Bloch, "Parallel Evolution of Image Processing Tools for Multispectral Imagery," *Proc. SPIE*, vol 4132, pp. 72–82, 2000.

[7] K.L. Hirsch, S.P. Brumby, N.R. Harvey and A.B. Davis, "The MTI Dense-Cloud Mask Algorithm Compared to a Cloud Mask Evolved by a Genetic Algorithm and to the MODIS Cloud Mask," *Proc. SPIE*, vol 4132, 2000.

[8] S.J. Perkins, J. Theiler, S.P. Brumby, N.R. Harvey, R.B. Porter, J.J. Szymanski and J.J. Bloch, "GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multispectral Images," *Proc. SPIE*, vol 4120, pp. 52–62, 2000.

[9] C. Harris and B. Buxton, "Evolving edge detectors," *Research Note RN/96/3, University College London, Dept. of Computer Science, London*, 1996.

[10] N.R. Harvey and S. Marshall, "GA Optimization of Multidimensional Grey-Scale Soft Morphological Filters with Applications in Archive Film Restoration," in *Mathematical Morphology and its Applications to Image and Signal Processing V (ISMM2000)*, J. Goutsias, L. Vincent, D.S. Bloomberg (Eds.), pp. 129–138, Norwell, MA: Kluwer Academic Publishers, 2000.

[11] A. Teller and M. Veloso, "A controlled experiment: Evolution for learning difficult image classification," *Lecture Notes in Computer Science*, vol. 990, pp. 165-176, 1995.

[12] R. Poli and S. Cagoni, "Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement," in *Genetic Programming 1997: Proc. 2nd Annual Conference on Genetic Programming* Koza, J.R., et al. (Eds.), San Francisco: Morgan Kaufmann, 1997.

[13] J.M. Daida, J.D. Hommes, T.F. Bersano-Begey, S.J. Ross and J.F. Vesecky, "Algorithm discovery using the genetic programming paradigm: Extracting low-contrast curvilinear features from SAR images of arctic ice," in *Advances in Genetic Programming 2, chap. 21*, P.J. Angeline, K.E. Kinnear, (Eds.), Cambridge, MA: MIT Press, 1996.

[14] S. Bandyopadhyay and S. K. Pal, "Pixel classification using variable string genetic algorithms with chromosome differentiation," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, pp. 303–308, 2001.

[15] L.A. Cox Jr., L. Davis and Y. Qiu, "Dynamic anticipatory routing in circuit-switched telecommunications networks," in *Handbook of Genetic Algorithms*, L. Davis (Ed.), pp. 124–143, New York: Van Nostrand Reinhold, 1991.

[16] T. Dandekar and P. Argos, "Potential of genetic algorithms in protein folding and protein engineering simulations," *Protein Engineering*, vol. 5 pp. 637–645, 1992.

[17] J.R. Koza, *Genetic Programming : On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)* Cambridge, MA: MIT Press, 1992.

[18] K.I. Laws, "Rapid texture identification," *Proc. SPIE*, vol. 238, pp. 376-380, 1980.

[19] M. Pietikainen, A. Rosenfeld and L.S. Davis, "Experiments with Texture Classification using Averages of Local Pattern Matches," *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-13, pp. 421–426. 1983.

[20] C.M. Bishop, *Neural Networks for Pattern Recognition*, pp. 105–112. Oxford University Press, 1995.

[21] `http://www.rsinc.com`

[22] `http://www.rsinc.com/Envi/tut2.cfm`

[23] J.A. Richards and X. Jia, "Remote Sensing Digital Image Analysis," New York: Springer-Verlag, 1999.

[24] F.A. Kruse, A.B. Lefkoff, J.B. Boardman, K.B. Heidebrecht, A.T. Shapiro, P.J. Barloon, and A.F.H. Goetz, "The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data," *Remote Sens. Environ.*, vol. 44, pp. 145–163, 1993.

[25] A.S. Mazer, M. Martin, M. Lee, J.E. Solomon, "Image Processing Software for Imaging Spectrometry Analysis," *Remote Sens. Environ.*, vol. 24, pp. 201–210, 1988.

[26] `http://aviris.jpl.nasa.gov`

[27] `http://aviris.jpl.nasa.gov/html/aviris.quicklooks.html`

[28] P.G. Weber, B.C. Brock, A.J. Garrett, B.W. Smith, C.C. Borel, W.B. Clodius, S.C. Bender, R.R. Kay, and M.L. Decker, "Multispectral Thermal Imager mission overview," *Proc. SPIE*, vol 3750, pp. 340–346, 1999.

[29] B. Jeon and D.A. Landgrebe, "Partially Supervised Classification Using Weighted Unsupervised Clustering," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, pp. 1073–1079, 1999.

[30] B. Shahshahani and D. Landgrebe, "Classification of Multi-Spectral Data by Joint Supervised-Unsupervised Learning," Technical Report TR-EE 94-1, School of Electrical Engineering, Purdue University, 1994.

[31] V. Castelli and T. M. Cover, "On the exponential value of labeled samples," *Pattern Recognition Lett.*, vol. 16, pp. 105–111, 1995.

[32] P.-F. Hsieh and D. Landgrebe, "Statistics enhancement in hyperspectral data analysis using spectral-spatial labeling, the EM algorithm, and the leave-one-out covariance estimator," *Proc. SPIE*, vol. 3438, pp. 183-190, 1999.

[33] L. Bruzzone and D. F. Prieto, "Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, pp. 456–460, 2001.

[34] J.T. Tou and R.C. Gonzales, *Pattern Recognition Principles*, Addison Wesley, 1974.

[35] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264–323, 1999.

[36] D. A. Landgrebe, "The development of a spectral-spatial classifier for Earth observational data," *Pattern Recognition*, vol. 12, pp. 165–175, 1980.

[37] B. Jeon and D. A. Landgrebe, "Classification with Spatio-Temporal Interpixel Class Dependency Contexts," *IEEE Trans. Geosci. Remote Sensing*, vol. 36, pp. 182–191, 1992.

[38] T. Yamazaki and D. Gingras, "Image Classification Using Spectral and Spatial Information Based on MRF Models," *IEEE Trans. Image Processing*, vol. 4, pp. 1333–1339, 1995.

[39] J. Theiler and G. Gisler, "A contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation," *Proc. SPIE*, vol. 3159, pp. 108–118, 1997.

Neal R. Harvey received a B.Eng. (Hons.) degree in Mechanical Engineering from the University of Hertfordshire, UK in 1989, a M.Sc. in Information Technology Systems from the University of Strathclyde, UK in 1992 and a Ph.D. in Non-linear Digital Image Processing from the University of Strathclyde in 1997. He spent over 6 years as a Research Fellow in the Signal Processing Division of the University of Strathclyde's Dept. of Electronic and Electrical Engineering. In April 1999, he took up a post-doctoral research associate position in the Space and Remote Sensing Sciences Group of the Nonproliferation and International Security Division at the Los Alamos National Laboratory, NM, USA, where he is currently employed. His research interests include nonlinear digital filters, optimization techniques, machine learning, image classification, remote sensing and film and video restoration.

Evolving forest fire burn severity classification algorithms for multi-spectral imagery

Steven P. Brumby, Neal R. Harvey, Jeffrey J. Bloch, James Theiler,
Simon Perkins, A. Cody Young, and John J. Szymanski

Space and Remote Sensing Sciences, Los Alamos National Laboratory,
Mail Stop D436, Los Alamos, New Mexico 87545, U.S.A.[*]

ABSTRACT

Between May 6 and May 18, 2000, the Cerro Grande/Los Alamos wildfire burned approximately 43,000 acres (17,500 ha) and 235 residences in the town of Los Alamos, NM. Initial estimates of forest damage included 17,000 acres (6,900 ha) of 70-100% tree mortality. Restoration efforts following the fire were complicated by the large scale of the fire, and by the presence of extensive natural and man-made hazards. These conditions forced a reliance on remote sensing techniques for mapping and classifying the burn region. During and after the fire, remote-sensing data was acquired from a variety of aircraft-based and satellite-based sensors, including Landsat 7. We now report on the application of a machine learning technique, implemented in a software package called GENIE, to the classification of forest fire burn severity using Landsat 7 ETM+ multispectral imagery. The details of this automatic classification are compared to the manually produced burn classification, which was derived from field observations and manual interpretation of high-resolution aerial color/infrared photography.

Keywords: Multispectral imagery, Genetic programming, Supervised classification, Forest fire, Wildfire.

1. INTRODUCTION: REMOTE SENSING OF FOREST FIRES

Between May 6 and May 18, 2000, the Cerro Grande/Los Alamos wildfire burned approximately 43,000 acres (17,500 ha) of forest and 235 residences in the town of Los Alamos, New Mexico (USA). Initial estimates of forest damage included 17,000 acres (6,900 ha) suffering 70-100% tree mortality. Some of the affected agencies and tribes included the United States Forest Service, the Department of Energy, the National Park Service, Santa Clara Pueblo, and the Pueblo of San Ildefonso. Restoration efforts following the fire were complicated by the large scale of the fire, and by the presence of extensive natural and man-made hazards. These conditions forced a reliance on remote sensing techniques for mapping and classifying the burn region. During and after the fire, remote-sensing data was acquired from a variety of aircraft-based and satellite-based sensors, including Landsat 7, to evaluate the impact of the fire.

Remote sensing of forest fires has traditionally involved human interpretation of visible wavelength and/or infrared photography. Since the introduction of aircraft and satellite mounted multi-spectral imaging instruments, e.g., the Advanced Very High Resolution Radiometer[1] (AVHRR) on the NOAA Polar-orbiting Operational Environmental Satellite (POES) series, and the Thematic Mapper (TM) and Enhanced Thematic Mapper (ETM+) instruments on the Landsat[2] series of Earth observation satellites, several physics-based and empirical algorithms for detecting forest fires have appeared in the literature. Two general approaches exist: detection of “hot-spots” and fire fronts, using, e.g., thresholds on brightness temperature[3,4,5,6,7] in AVHRR band 3 (3.7µm), and mapping of post-fire burn scars. Landsat 7 fortuitously captured an image of the Cerro Grande/Los Alamos wildfire in progress on May 9, 2000 (Landsat Path 33, Row 35), in which fire fronts due to the wildfire and the back-burning efforts of the fire fighters are clearly visible (Fig. 1).

For the present work, however, we are interested in mapping and classifying the post-fire burn scar. A number of researchers have investigated the use of Landsat TM imagery for measuring wildfire impact by mapping of the burn scar. For example, Lobo et al[8] apply a combination of spectral image segmentation and hierarchical clustering to the mapping and analysis of

[*] Work supported by U.S. Departments of Energy and Defense. Further author information: (Send correspondence to S.P.B.)
Emails: `{brumby,jbloch,harve,s.perkins,jt,cody,szymanski}@lanl.gov`

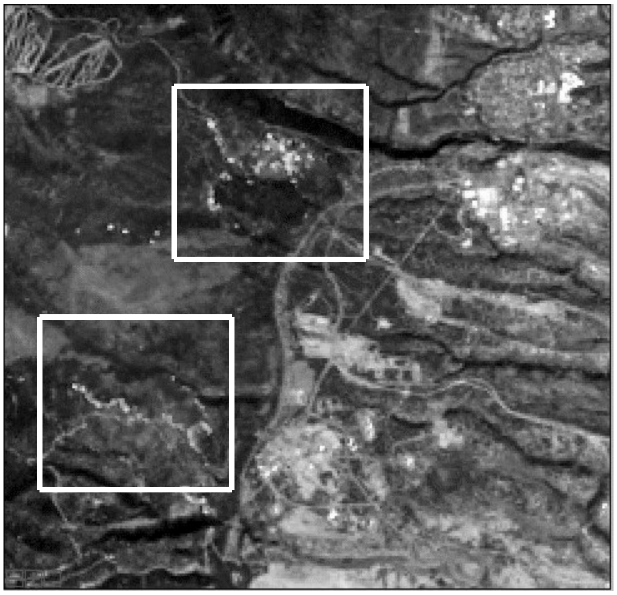


Figure 1. Wildfire hotspots, May 9, 2000: Bright pixels within the white boxes are wildfire hotspots. The town and laboratory of Los Alamos lie along the right edge of this image. Pixel size: 30m, ETM+ bands 7,5,3.

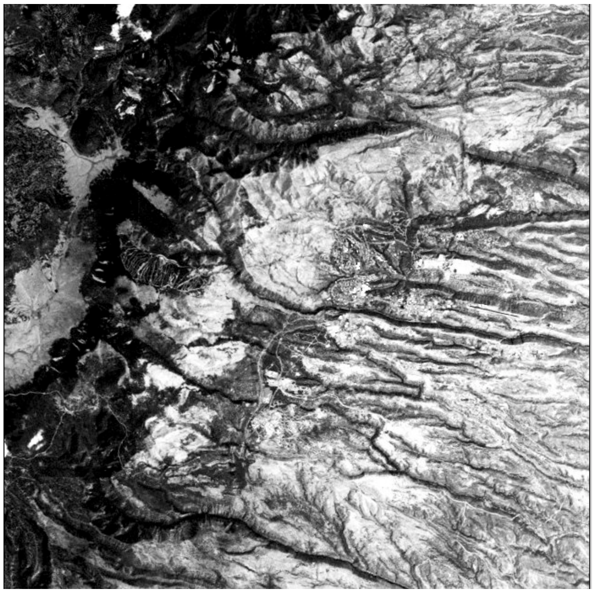


Figure 2. Post-fire, July 19, 2000: Bright region in center of image is the burn scar. Los Alamos town lies against the underside of the burn scar. Topography changes from forested mountains (left) to bare mesas.

fires in Mediterranean forests. Kushla and Ripple[9] use Landsat imagery to map forest survival following a wildfire in western Oregon (USA), and investigate linear combinations of post-fire and multi-temporal TM band ratios and differences.

We now report on the application of a machine learning technique to the classification of forest fire burn severity using Landsat 7 ETM+ multispectral imagery. The details of this automatic classification are compared to a manually produced burn classification, which was derived from field observations and human photo-interpretation of high-resolution aerial color/infrared photography.

2. MACHINE LEARNING: GENETIC ALGORITHM + SUPERVISED CLASSIFIER

GENIE[10,11,12,13] is an evolutionary computation (EC) software system, using a genetic algorithm[14,15,16] (GA) to assemble image-processing algorithms from a collection of low-level ("primitive") image processing operators (e.g., edge detectors, texture measures, spectral operations, and various morphological filters). This system has been shown to be effective in looking for complex terrain features, such as, e.g., golf courses[17]. GENIE can sequentially extract multiple features for the same scene to produce terrain classifications[18]. GENIE has been described at length elsewhere (see previous references), so we will only present a brief description of the system here.

GENIE follows the classic evolutionary paradigm: a population of candidate image-processing algorithms is randomly generated, and the fitness of each individual assessed from its performance in its environment, which for our case is a user-provided training scene. After fitness has been assigned, reproduction with modification follows via the evolutionary operators of selection, crossover, and mutation, applied to the most fit members of the population. The process of fitness evaluation and reproduction with modification is iterated until some stopping condition is satisfied.

The algorithms assembled by GENIE will generally combine spatial and spectral processing, and the system was in fact designed to enable spatio-spectral image processing experimentation. Each individual chromosome in the population consists of a fixed-length string of genes. Each gene in Genie corresponds to a primitive image processing operation, and so the whole chromosome describes an algorithm consisting of a sequence of primitive image processing steps. We now briefly describe our method of providing training data, our encoding of image-processing algorithms as chromosomes for manipulation by the GA, and our method for evaluating the fitness of individuals in the population.

2.1. Training Data

The environment for the population consists of one or a number of training scenes. Each training scene contains a raw multi-spectral image data cube, together with a weight plane and a truth plane. The weight plane identifies the pixels to be used in training, and the truth plane locates the features of interest in the training data. Providing sufficient quantities of good training data is a crucial to the success of any machine learning technique. In principle, the weight and truth planes may be derived from an actual ground campaign (i.e., collected on the ground at the time the image was taken), may be the result of applying some existing algorithm, and/or may be marked-up by hand using the best judgement of an analyst looking at the data. We have developed a graphical user interface (GUI), called Aladdin, for manual marking up of raw imagery. Using Aladdin, the analyst can view a multi-spectra image in a variety of ways, and can mark up training data by painting directly on the image using the mouse. Training data is ternary-valued, with the possible values being "true", "false", and "unknown". True defines areas where the analyst is confident that the feature of interest does exist. False defines areas where the analyst is confident that the feature of interest does not exist. Unknown pixels do not influence the fitness of a candidate algorithm.

2.2. Representation of Image-Processing Algorithms

Traditional genetic programming[19] (GP) uses a variable sized (within limits) tree representation for algorithms. Our representation differs in that it allows for reuse of values computed by sub-trees, i.e. the resulting algorithm is a graph rather than a tree. The image processing algorithm that a given chromosome represents can be thought of as a directed acyclic graph where the non-terminal nodes are primitive image processing operations, and the terminal nodes are individual image planes extracted from the multi-spectral image used as input. Our representation also differs in that the total number of nodes is fixed (although not all of these may actually be used in the final graph), and crossover is carried out directly on the linear representation.

We have restricted our "gene pool" to a set of useful primitive image processing operators ("genes"). These include spectral, spatial, logical and thresholding operators. The set of morphological operators is restricted to function-set processing morphological operators, i.e., gray-scale morphological operators having a flat structuring element. The sizes and shapes of the structuring elements used by these operators is also restricted to a pre-defined set of primitive shapes, which includes the square, circle, diamond, horizontal cross and diagonal cross, and horizontal, diagonal, and vertical lines. The shape and size of the structuring element are defined by operator parameters. Other local neighborhood/windowing operators such as mean, median, etc., specify their kernels/windows in a similar way. The spectral operators have been chosen to permit weighted sums, differences and ratios of data and/or "scratch" planes, where a scratch plane is a block of memory for storing intermediate calculations within a candidate image-processing algorithm.

A single gene consists of an operator, plus a variable number of input arguments specifying from where input is read, output arguments specifying where output is to be written, and any additional parameters that might be required to specify how the specific operator works (e.g., the diameter and shape of a structuring element used in a morphological filter). The operators used in Genie take one or more distinct image planes as input, and generally produce a single image plane as output. Input can be taken from any data plane in the training data image cube. Output is written to one of a number of scratch planes, temporary workspaces where an image plane can be stored. Genes can also take input from scratch planes, but only if that scratch plane has been written to by another gene positioned earlier in the chromosome sequence. We use a notation for genes[10] that is most easily illustrated by an example: the gene `[ADDP rD0 rS1 wS2]` applies pixel-by-pixel addition to two input planes, read from data plane 0 and from scratch plane 1, and writes its output to scratch plane 2. Any additional required operator parameters are listed after the output arguments.

Note that although all chromosomes have the same fixed number of genes, the effective length of the resulting algorithm may be smaller than this. For instance, an operator may write to a scratch plane that is then overwritten by another gene before anything reads from it. GENIE performs an analysis of chromosome graphs when they are created and only carries out those processing steps that actually affect the final result. Therefore, the fixed length of the chromosome acts as a maximum effective length.

2.3. Supervised Classification and Fitness Evaluation

Each candidate image-processing algorithm generates a number of intermediate feature planes (or "signature" planes), which are then combined to generate a Boolean-valued mask for the feature of interest. This combination is achieved using a standard supervised classifier (we use the Fisher linear discriminant[20]), and an optimal threshold function.

Complete (or "hard") classification requires that the image-processing algorithm produce a binary-valued output plane for any given scene. It is possible to treat, e.g., the contents of the first scratch plane as the final output for that candidate image-processing algorithm (thresholding would generally be required to obtain a binary result, though Genie can choose to apply its own Boolean thresholding functions). However, we have found it to be useful to perform the combination of the data and scratch planes using a non-evolutionary method, and have implemented a supervised classifier backend. To do this, we first select a subset of the scratch planes and data planes to be "signature" planes. For the present experiments, this subset consists of just the scratch planes. We then use the provided training data and the contents of the signature planes to derive the Fisher Discriminant, which is the linear combination of the signature planes that maximizes the mean separation in spectral terms between those pixels marked up as "true "and those pixels marked up as "false", normalized by the total variance in the projection defined by the linear combination. The output of the discriminant-finding phase is a real-valued single-plane "answer" image. This is reduced to a binary image by exhaustive search over all the training pixels to find the threshold value that minimizes the total number of misclassifications (false positives plus false negatives) on the training data.

The fitness of a candidate solution is given by the degree of agreement between the final binary output plane and the training data. This degree of agreement is determined by the Hamming distance between the final binary output of the algorithm and the training data, with only pixels marked as true or false (as recorded in the weight plane) contributing towards the metric. The Hamming distance is then normalized so that a perfect score is 1000.

2.4. Software Implementation

GENIE can search a rich and complex feature space using its gene pool of standard primitive image processing operators, and the results of additional analyst-selected algorithms. The system employs both spectral and spatial image analysis techniques in combination, and can in principal simultaneously exploit data from different sensors (e.g., optical imagery plus multi-spectral imagery plus altimeter data or digital elevation models). The ability to combine diverse datasets requires that the data be co-registered, which requires use of some other package (e.g., RSI's ENVI[21] or ERDAS's Imagine[22] software packages).

Our genetic algorithm code has been implemented in object-oriented Perl. This provides a convenient environment for the string manipulations required by the evolutionary operations, and easy access to the underlying operating system (Linux). Chromosome fitness evaluation is the computationally intensive part of the evolutionary process, typically taking 90% of our total processing time. We currently use RSI's IDL[21] language and image processing environment for this core processing, because of its visualization environment, and its ability to handle a diverse set of imagery formats. Within IDL, individual genes correspond to single primitive image operators, which are coded as IDL procedures. A chromosome can then be represented as an IDL batch executable. Many of our primitive operators do not exist in standard IDL, so we have developed an external library of C code called by IDL. In the present implementation, an IDL session is opened at the start of a run and communicates with the Perl code via a two-way UNIX pipe. This pipe is a low-bandwidth connection. Only the IDL session needs to access the input and training data (possibly hundreds of Megabytes), which requires a high-bandwidth connection. The Aladdin training data mark-up application was written in Java. Running on a single, fast Linux/Intel workstation, the system typically requires a few hours to evolve an image-processing algorithm. Re-application of an evolved image-processing algorithm to the same or a new image typically takes seconds to minutes.

3. TRAINING AND RESULTS

3.1. Training Data

The remotely-sensed images used in this paper are Landsat 7 ETM+ 30 meter multi-spectral data (ETM+ bands 1–5 and 7). These scenes are Level 1G radiance corrected and georeferenced standard data products obtained via the U.S. Geological Survey (USGS) EarthExplorer[23] web site. We used a post-fire Landsat scene from July 17, 2000, Path 34 and Row 35. The image displayed in Fig. 2 is a false-color image, which has then been converted to gray-scale and has had its contrast enhanced for the printing process. As we are interested in mapping burn scars, we generally view the data using a Visible/Infrared/Thermal pattern of a thermal IR band (ETM + band 7, 2.2μm) for the red component, a near IR band for the green component (band 5, 1.65μm), and a visible red band for the blue component (band 3, 0.66μm). A Landsat 7 Path/Row swath has an across-track field-of-view of approximately 185 km, with similar along-track length, resulting in a field-of-view of approximately 34,000 sq.km, which is much larger than needed for this study, and presents memory problems for our software if we attempt to ingest the whole scene. Hence, we spatially subset the image to a 1000 pixel x 1000 pixel region centered on the Los Alamos National Laboratory. We chose not to use the 60m thermal or 15m panchromatic data in the following analysis, as we wished to investigate evolution without the added complication of re-sampling of data.

We did not have any atmospheric measurements available for the scene, so we did not attempt to carry out any corrections for haze or atmosphere. The topography of Los Alamos is complex, consisting of a dormant volcano (the Jemez Mountains) rising to approximately 10,000 feet (3.3km), surrounded by a radiating network of mesas at 7,000 – 8,000 feet, falling off to the Rio Grande river valley at approximately 6,500 feet elevation. Traditionally, illumination effects due to complex

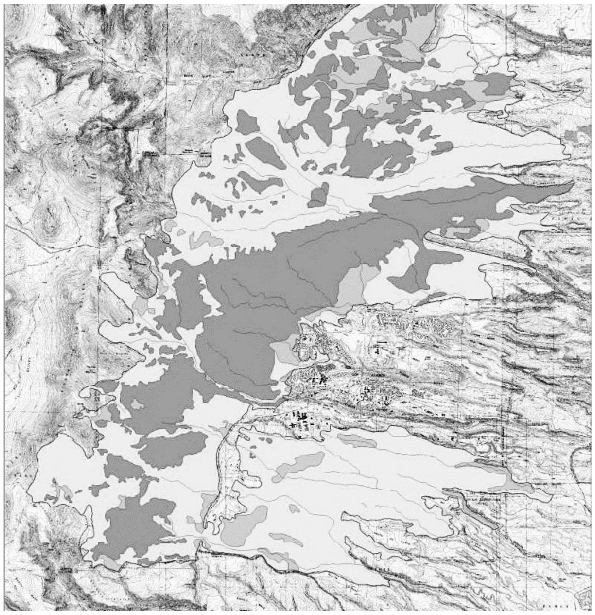


Figure 3. BAER Team burn-severity map over topographic map: Medium gray region marks high severity burn, pale gray region marks low severity/unburned region: http://www.baerteam.org/cerrogrande

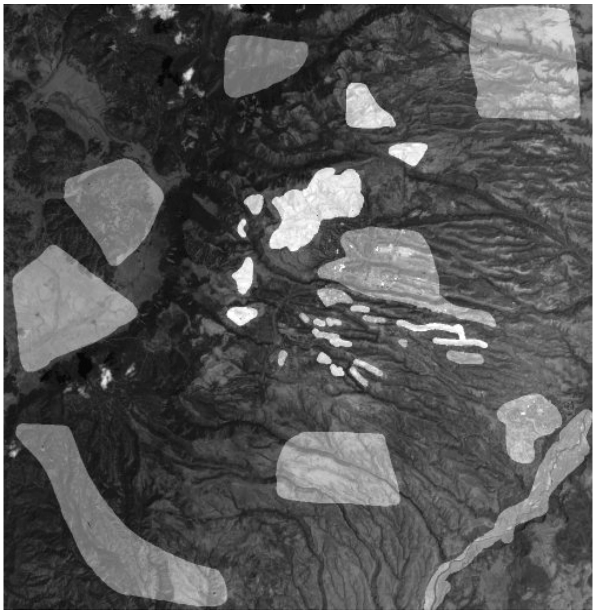


Figure 4. Training Data over raw imagery: White patches mark "burn" regions. Gray patches mark "non-burn" regions. Note: this image is presented at a larger spatial scale than Figure 3.

topography can be approximately "factored out" by using band ratios, or removed using principal components analysis (see, e.g., Ref. 24). Here, we are interested in the GENIE software's ability to derive results based on the raw imagery, and do not add any additional band ratio or band difference planes.

Our training data was based on the official Cerro Grande Burned-Area Emergency Rehabilitation (BAER) Team's burn severity map, Figure 3, which was produced by trained observers flying over the fire, and visual inspection of high-resolution (~1 meter) aerial color/infrared photography collected during and immediately after the fire. Using this map as a guide, we marked up several regions of the Landsat image as almost certainly "burn", and several regions as almost certainly "non-burn", as shown in Figure 4. The BAER Team assign "burn severity" on the basis of tree mortality – low burn severity corresponds to grass fire and low tree mortality, medium severity burn classification implies crown fire and majority tree mortality (more than half of the trees in the marked region are dead), and the high severity burn classification requires that 70 – 100% of the trees are dead. The Cerro Grande wildfire tended to produce either high severity or low severity burn, with only a relatively small fraction of the burn classified as medium burn severity in the BAER Team maps. This was mostly due to the over-grown nature of the Ponderosa pine/mixed conifer forest which suffered most of the damage. Major species present include Pinus ponderosa (Ponderosa Pine), Pseudotsuga menzii (Douglas fir), Abies concolor (White fir), Populus tremuloides (Aspen), Juniperus monosperma (Juniper), and Pinus edulis (Piñon). Note that we have also tried evolving algorithms from training data based purely on photo-interpretation of the 30m Landsat scene, and have obtained similar results. This is almost certainly due to the fact that in the case of the Cerro Grande wildfire the burn damage was sufficiently catastrophic that simple inspection of the 30m imagery allows accurate marking of "burn" and "non-burn" regions

3.2. Evolved Image-Processing Algorithm

The system was run for with a population of 50 chromosomes, each having a fixed length of 20 genes, and 3 intermediate feature ("scratch") planes. The GA was allowed to evolve for 30 generations, in this case, evaluating 1282 distinct candidate image processing algorithms, which is very small compared to search space of possible algorithms given our representation. This required approximately 7 hours of wall-clock time running on a 500MHz Linux/Intel Pentium 2 workstation.

The best evolved image-processing algorithm had the chromosome,

```
[OPEN rD1 wS1 1 1][ADDS rD4 wS3 0.34][NEG rS1 wS1][MULTP rD4 rS3 wS2]
[LINCOMB rS1 rD6 wS3 0.11][ADDP rS1 rS3 wS1][SUBP rS1 rD5 wS1]
```

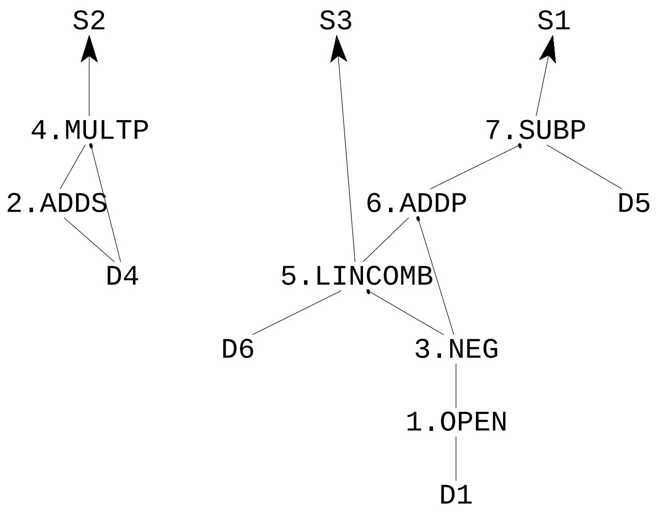


Figure 5. Evolved image-processing algorithm: each node (gene) in the graph is labeled by its position along the length of the chromosome and by the GENIE software's mnemonic for the primitive image processing operator.

and is shown as a diagram in Figure 5, where each node (gene) in the graph is labeled by its position along the length of the chromosome and by our mnemonic for the operator (e.g., 1. OPEN for a morphological opening operator). In words, the image-processing algorithm works as follows. Note that GENIE converts the byte-valued raw data to real-valued data (64 bit doubles) and keeps that precision through all its calculations.

1. Data plane D1 (ETM+ band 1, visible blue 0.48μm) undergoes a grayscale morphological opening operation (node 1. OPEN) using a "circular" structuring element with diameter equal to 3 pixels (equivalent to a 3x3 square with corners removed) and the result is written to scratch plane S1,
2. The negative of this plane is taken (node 3. NEG), i.e., S1 → – S1,
3. The new S1 is linearly combined (node 5. LINCOMB) with data plane D6 (ETM+ band 7, medium wavelength infrared (MWIR) 2.22μm) with linear weights: 0.11*S1 + 0.89*D6 and the result written to scratch plane S3 (its final value),
4. Scratch planes S1 and S3 are summed (node 6. ADDP), and the difference (node 7. SUBP) of this sum and data plane D5 (ETM+ band 5, MWIR 1.65μm), S1 + S3 – D5, is written to S1 (its final value),
5. Data plane D4 (ETM+ band 4, near infrared 0.83μm) has a constant, 0.34 times a DATASCALE variable equal to the range of the input raw data values, added to each pixel (node 2. ADDS) and is multiplied by D4 again to form the linear combination D4*D4 + (0.34*DATASCALE)*D4, which is written to scratch plane S2 (its final value).

The final values of S1, S2, and S3 are then combined in the linear sum, where the coefficients and intercept have been chosen by the Fisher discriminant, as described in Section 2.3, above, to produce our real-valued answer plane A (Figure 6):

$$A = 0.0147*S1 - 0.0142*S2 + 0.0134*S3 + 1.554$$

The optimal threshold found by GENIE, given the training data, was 0.3437. Converting A to a Boolean mask at that threshold value produces Figure 7. In relation to the BAER map, Figure 3, we see that the system has extracted the high, medium, and low severity burn regions, but also presents a number of false positives. On inspection, these turn out to correspond to two physical categories of land cover: bare ground/rock, and cloud shadows. The histogram of A shows a bimodal distribution (Figure 8), as expected if the burn/non-burn classes are separable. Adjusting the threshold on A to fall at the between-peak minimum of the histogram at 0.7930 (a different optimization criterion for the threshold than that used by default by GENIE) produces a new Boolean mask, Figure 9, in which almost all the false positives have been removed, and the remaining pixels marked as "burn" correspond very closely to the high severity burn regions in the BAER map

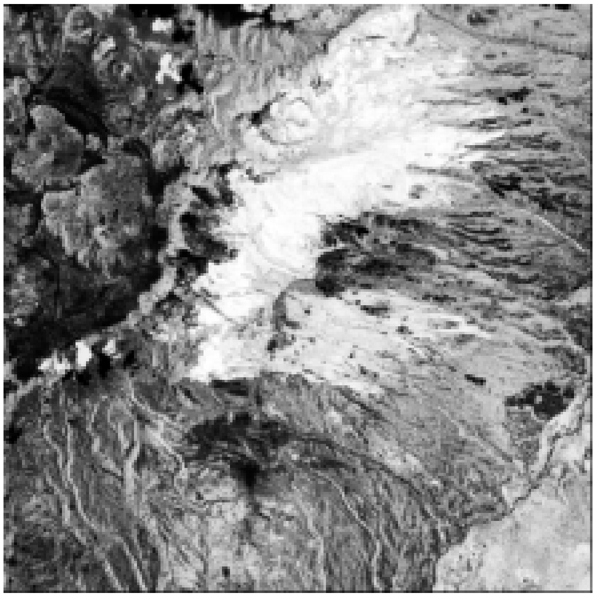


Figure 6. Real-valued Answer Plane: We use a Fisher Discriminant to find the optimal linear combination of evolved "signature" planes into a real-valued answer plane. Regions which will tend to be classified as "burn" are bright. This image has been histogram-equalized to increase contrast.

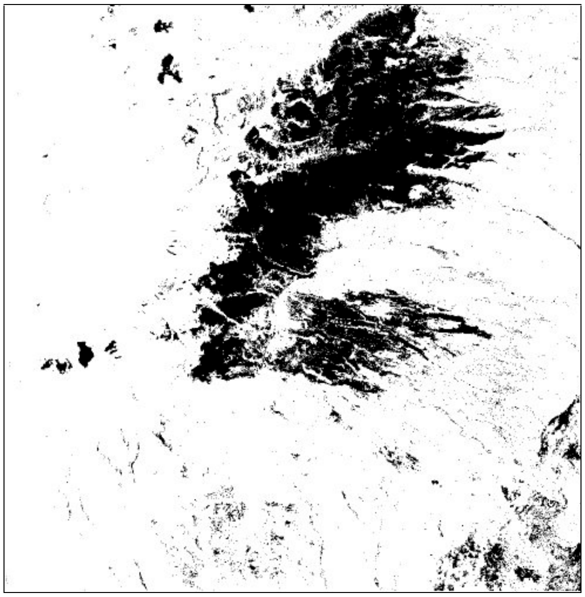


Figure 7. Burn mask: GENIE determines an optimal threshold for converting the real-valued answer plane to a Boolean mask. Misidentified pixels are mostly cloud shadows (e.g., compact regions on left), or bare ground/rock (lower right and bottom).

3.3. Application to Non-Training Data

The evolved algorithm can now be applied to any scene. To check the reasonableness of our algorithm's performance, we ran the image-processing algorithm over a larger fraction of the Landsat scene, encompassing the entire Jemez mountain range. The result is shown in Figure 10. We claim that this overall result is quite reasonable, and only fails where the Landsat swath ends (which can be easily masked out). Of particular interest is the persistent detection of a severe burn site on the Western side of the Jemez mountains, Figures 10, 11, 12, which cannot obviously be excluded due to cloud shadows or data drop-out. In fact, this turns out to be a true detection of a second wildfire, the Stable wildfire (effecting Stable Stream and School House Mesa in the Jemez Mountains of northern New Mexico), which destroyed approximately 800 acres of forest in September/October of 1999. As GENIE had no knowledge of this fire during its training, we find this detection, together with the reasonable behavior of the evolved image-processing algorithm over this large region, as quite encouraging for the future usefulness of this machine learning technique.

4. CONCLUSIONS

We have investigated evolution of an image-processing algorithm to extract wildfire burn scars in Landsat 7 ETM+ imagery, and have described the operation of the evolved algorithm in some detail. The evolved algorithm shows a good qualitative fit to the published BAER Team burn-severity map of the May 2000 Cerro Grande/Los Alamos wildfire, specifically in comparison to their high-severity burn class (70-100% tree mortality regions). The algorithm can be confused by dark cloud shadows, and by bare ground/rock outcrops which are physically very similar to the charred remains of the severely burned forest, but adjustment of its final threshold can significantly improve this behavior. Applying the algorithm outside the training area showed that it continued to produce reasonable results over a large spatial region, and in fact was able to detect a second small wildfire on the west side of the Jemez mountains (September/October 1999 Stable wildfire). We find these results quite encouraging for the future application of this machine learning technique.

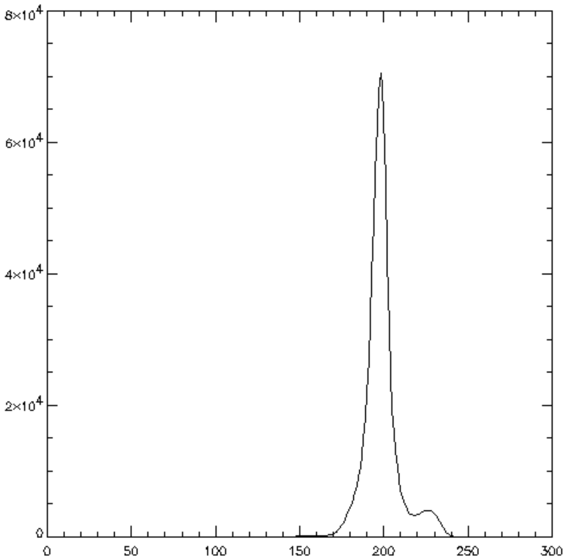


Figure 8. Histogram of the Answer Plane (Fig. 6): The bimodal distribution indicates that **"**burn" and "non-burn" are indeed separable classes.

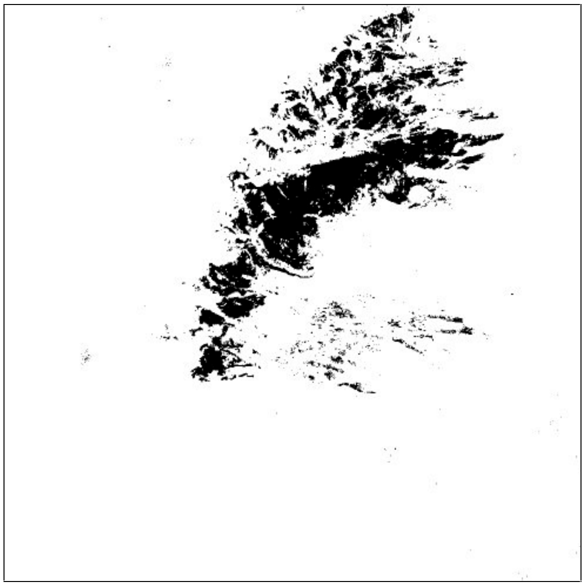


Figure 9. Final burn mask: Thresholding the answer plane at the between-peak minimum of the bimodal distribution produces this burn mask, which has almost eliminated false positives. There is substantial agreement with the details of the BAER map (Fig.3).

ACKNOWLEDGEMENTS

The authors would like to thank Leslie Hanson, Steven Koch, and Randy Balice of the Ecology Group for useful discussions and access to data used in this work. The GENIE system is the result of the combined efforts of a team of people at LANL, including, in addition to the authors of this paper: Reid Porter, Mark Galassi, Kevin Lacker, and Melanie Mitchell.

REFERENCES

1. The National Oceanographic and Atmospheric Administration (NOAA) POES satellites and the AVHRR instrument are described on the NOAA web site http://www.ncdc.noaa.gov
2. Landsat TM and ETM+ are described on the U.S. Geological Survey (USGS) web site http://landsat7.usgs.gov
3. Y.J. Kaufmann, C.J. Tucker, and I. Fung, "Remote sensing of biomass burning in the tropics", J. Geophysical Research, Vol 95, No. D7, pp. 9927-9939, 1990, and references therein.
4. Y. Rauste, E. Herland, H. Frelander, K. Soini, T. Kuoremäki, and A. Ruokari, "Satellite-based forest fire detection for fire control in boreal forests", Int. J. Remote Sensing, Vol. 18, No. 12, pp. 2641-2656, 1997.
5. N.P. Minko, N.A. Abushenko, V.V. Koshelev, "Forest fire detection in East Siberia forests using AVHRR/NOAA data", Proc. SPIE, Vol. 3502, pp. 192-200, 1998.
6. R. Lasaponara, V. Cuomo, V. Tramutoli, N. Pergola, C. Pietrapertosa, and T. Simoniello, "Forest fire danger estimation based on the integration of satellite AVHRR data and topographic factors", Proc. SPIE, Vol. 3868, pp. 241-252, 1999.
7. S.H. Boles and D.L. Verbyla, "Effect of scan angle on AVHRR fire detection accuracy in interior Alaska", Int. J. Remote Sensing, Vol. 20, No. 17, 3437-3443, 1999.
8. A. Lobo, N. Pineda, R. Navarro-Cedillo, P. Fernandez-Rebollo, F.J. Salas, J.-L. Fernández-Turiel, and A. Fernández-Palacios, "Mapping forest fire impact from Landsat TM imagery", Proc. SPIE, Vol. 3499, 340-347, 1998.
9. J.D. Kushla and W.J. Ripple, "Assessing wildfire effects with Landsat thematic mapper data", Int. J. Remote Sensing, Vol. 19, No. 13, 2493-2507.
10. S.P. Brumby, J. Theiler, S.J. Perkins, N.R. Harvey, J.J. Szymanski, J.J. Bloch, and M. Mitchell, "Investigation of feature extraction by a genetic algorithm ", Proc. SPIE, Vol. 3812, pp. 24-31,1999.
11. J. Theiler, N.R. Harvey, S.P. Brumby, J.J. Szymanski, S. Alferink, S.J. Perkins, R. Porter, and J.J. Bloch, "Evolving retrieval algorithms with a genetic programming scheme ", Proc. SPIE, Vol. 3753, pp.416-425,1999.

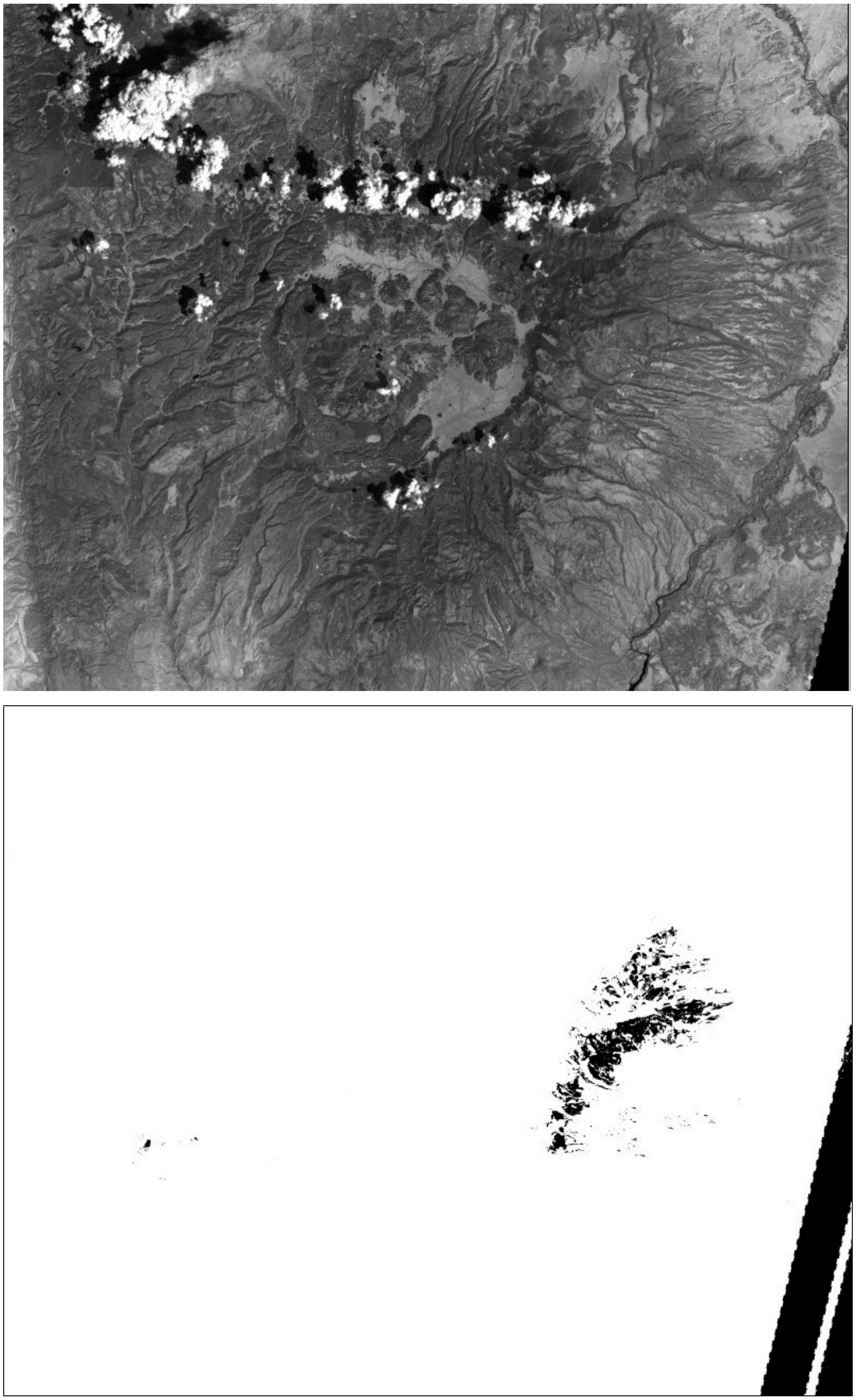


Figure 10 Testing the evolved image-processing algorithm: Extended region (top) and burn mask (bottom). The evolved image-processing algorithm continues to work well, except at the edge of the Landsat swath (bottom image, lower right). The small black region on the left of the burn mask represents a true detection of a second recent wildfire, the 1999 Stable wildfire.

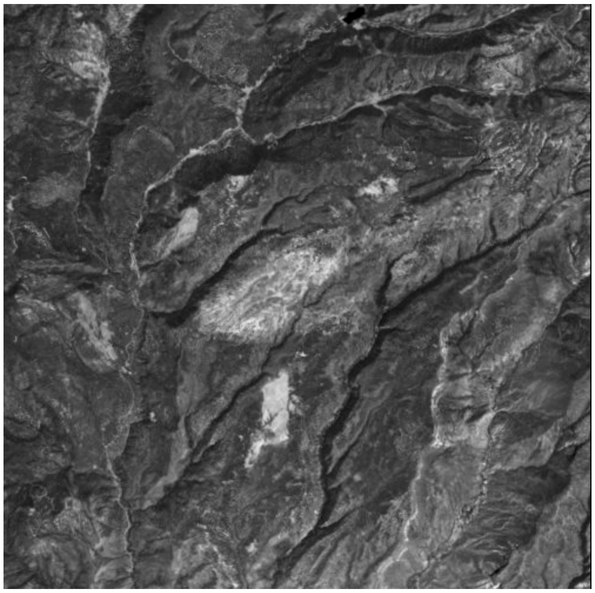


Figure 11. Detail of the second detected burn: Grayscale image, ETM+ band 7.

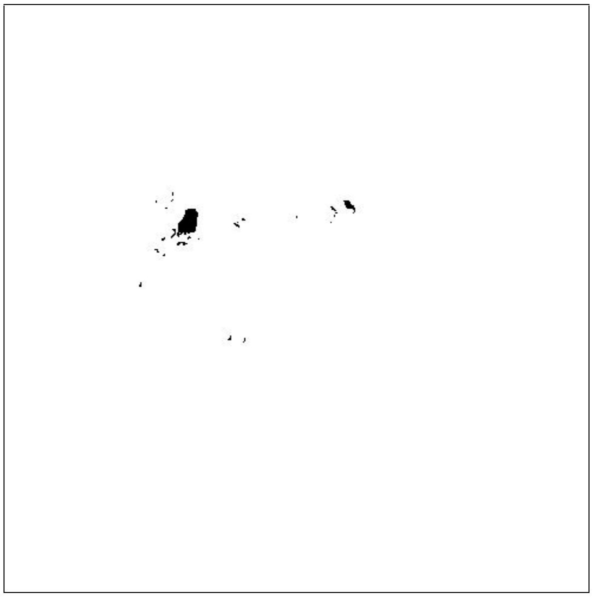


Figure 12. Burn mask for second detected burn: Location agrees with the known 1999 Stable wildfire.

12. N.R. Harvey, S.P. Brumby, S.J. Perkins, R.B. Porter, J. Theiler, A.C. Young, J.J. Szymanski, and J.J. Bloch, “Parallel evolution of image processing tools for multispectral imagery”, Proc. SPIE, Vol. 4132, pp. 72-82, 2000.
13. S. Perkins, J. Theiler, S.P. Brumby, N.R. Harvey, R.B. Porter, J.J.Szymanski, and J. J. Bloch, “GENIE: A hybrid genetic algorithm for feature classification in multi-spectral images”, Proc. SPIE, Vol. 4120, pp 52-62, 2000.
14. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan, Ann Arbor (1975).
15. I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Fromman-Holzboog*, Stuttgart (1973).
16. L. Fogel, A. Owens and M. Walsh, *Artificial Intelligence through Simulated Evolution*, Wiley, New York (1966).
17. N.R. Harvey, S. Perkins, S.P. Brumby, J. Theiler, R.B. Porter, A.C. Young, A.K. Varghese, J.J. Szymanski, and J.J. Bloch, “Finding golf courses: The ultra high tech approach”, Proc. Second European Workshop on Evolutionary Computation in Image Analysis and Signal Processing (EvoIASP2000), Edinburgh, UK, pp 54-64, 2000.
18. S.P. Brumby, N.R. Harvey, S. Perkins, R.B. Porter, J.J. Szymanski, J. Theiler, and J.J. Bloch, “A genetic algorithm for combining new and existing image processing tools for multispectral imagery”, Proc. SPIE, Vol. 4049, pp. 480-490, 2000.
19. J. R. Koza, *Genetic Programming: On the Programming of Computers by Natural Selection*, MIT, Cambridge (1992).
20. For example, see C.M.Bishop, *Neural Networks for Pattern Recognition*, pp.105 –112, Oxford University (1995).
21. For details on RSI Inc.’s ENVI and IDL software, see http://www.rsinc.com
22. For details on ERDAS Inc.’s Imagine software, see http://www.erdas.com
23. For details on USGS EarthExplorer, see http://edcsns17.cr.usgs.gov/EarthExplorer.
24. R.A. Schowengerdt, *Remote Sensin*g, 2nd ed., Academic, San Diego (1997).

GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multi-Spectral Images

Simon Perkins, James Theiler, Steven P. Brumby, Neal R. Harvey, Reid Porter,
John J. Szymanski and Jeffrey J. Bloch

Space and Remote Sensing Sciences, Los Alamos National Laboratory,
Los Alamos, NM 87545, USA

ABSTRACT

We consider the problem of pixel-by-pixel classification of a multi-spectral image using supervised learning. Conventional supervised classification techniques such as maximiun likelihood classifcation and less conventional ones such as neural networks, typically base such classifications solely on the spectral components of each pixel. It is easy to see why: the color of a pixel provides a nice, bounded, fixed dimensional space in which these classifiers work well.

It is often the case however, that spectral information alone is not sufficient to correctly classify a pixel. Maybe spatial neighborhood information is required as well. Or maybe the raw spectral components do not themselves make for easy classification, but some arithmetic combination of them would. In either of these cases we have the problem of selecting suitable spatial, spectral or spatio-spectral features that allow the classifier to do its job well. The number of all possible such features is extremely large. How can we select a suitable subset?

We have developed GENIE, a hybrid learning system that combines a genetic algorithm that searches a space of image processing operations for a set that can produce suitable feature planes, and a more conventional classifier which uses those feature planes to output a final classification.

In this paper we show that the use of a hybrid GA provides significant advantages over using either a GA alone or more conventional classification methods alone. We present results using high-resolution IKONOS data, looking for regions of burned forest and for roads.

Keywords: Genetic algorithms, genetic programming, hybrid genetic algorithms, image feature classification, remote sensing, spatial context

1. INTRODUCTION

Modern satellite and aerial remote-sensing instruments produce ever greater quantities of image data of the Earth, at higher resolutions and with more spectral channels than ever before. Making sense of this data within a reasonable time-frame requires automated systems that can quickly and reliably interpret this data and extract information of interest to analysts.

Our particular area of concern is in performing pixel-by-pixel classifications of multi-spectral remotely-sensed images, producing overlays that show the locations of features of interest. The range of features we look for is very large, ranging from broad-area features such as forest and open water, through to man-made features such as roads and buildings. The broad range of features we're looking for, and the variety of instruments that we work with, make hand coding feature-finders impractical. Therefore we use a supervised learning scheme that, starting from a few hand-classified images, can automatically develop image processing pipelines that can distinguish the feature of interest from non-features. As an added bonus, it is often much quicker to develop a feature-detector automatically in this way, compared with writing the detector by hand, and so our system is very useful when a new feature must be located quickly in a large data set.

Supervised learning applied to remote-sensed images is not a new field, and many different statistical and machine learning techniques have been tried, ranging from maximum likelihood classifiers through to neural networks. Chapter 8 of Richards and Jia[1] provides a good summary.

The vast majority of supervised learning applied to remote-sensed imagery bases classification purely upon the feature vectors formed by the set of intensity values in each spectral channel for each pixel. This vector of numbers

E-mail contact: `s.perkins@lanl.gov`

provides a nice fixed-dimensionality multi-dimensional space in which conventional classifiers work well. However, it is often the case that spectral information alone is insufficient to correctly identify a pixel — often some feature of its neighborhood, e.g. texture, or the average value of nearby pixels, is necessary to disambiguate the spectral information. We can imagine that many different kinds of extra spatial context information could be added into the pixel feature vector as additional feature dimensions. Now consider that even in the spectral domain it may be that the raw spectral intensity values do not make for an easy classification. For instance, perhaps the normalized vector of spectral intensities would be easier to work with, and provide better robustness with respect to varying illumination. It is easy to see that there are a large number of choices for additional feature vector dimensions, and easy to see that the correct choice could make classification much easier than just taking the raw spectral values as our feature vector. The question is: how do we choose a suitable set of features automatically?

We have developed a hybrid evolutionary algorithm called Genie to do just this. The essential idea behind Genie is that a genetic algorithm[2] searches a very large space of image processing algorithms that transform raw multi-spectral pixel data into a new set of image planes that we call *feature planes.* A conventional supervised classifier is then applied to these feature planes and outputs the final classification. It is our experience that this hybrid combination of evolutionary search and conventional classification techniques can be very powerful.

Previous work[3–5] has described some of the empirical results achieved with Genie. This paper concentrates on the hybrid aspects of the system, and in particular compares the performance of the hybrid system with a pure evolutionary system and with a pure conventional classifier.

2. THE GENIE SYSTEM

2.1. Training Data

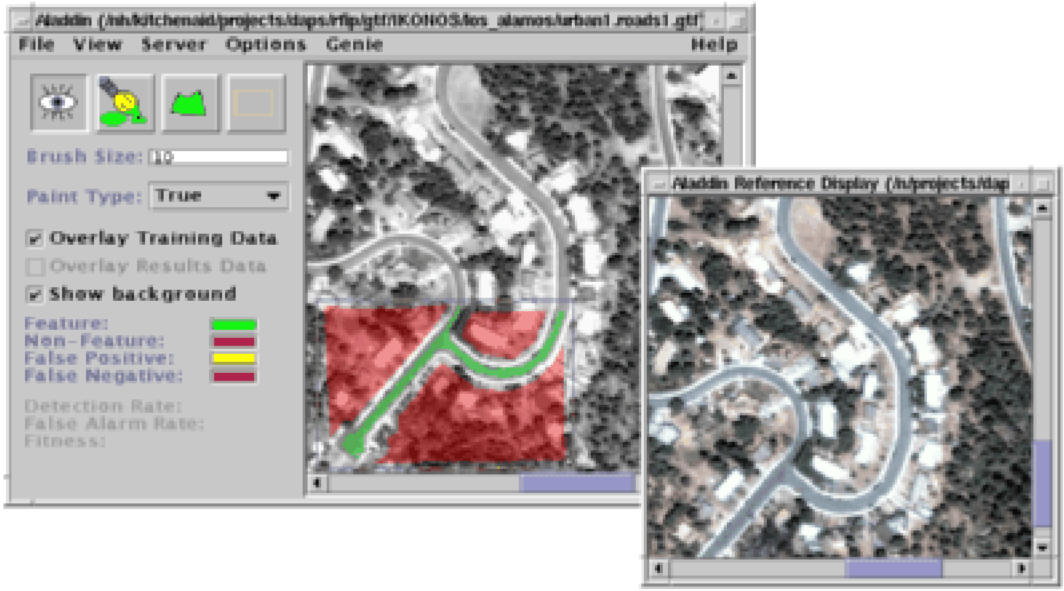
In GENIE, we typically assume that training data is provided by a human expert analyst, marking up an image by hand, showing both locations of the feature of interest, and locations where that feature is definitely not found. We have developed a Java-based tool called Aladdin which tries to make this process as painless as possible. Aladdin provides a 'paint program' environment in which an analyst can simultaneously view a false color projection of a multi-spectral image, and a grayscale version of the same image on which he or she can 'paint' training data as a colored overlay. Aladdin provides many user-friendly features to make this job easier, such as the ability to zoom into an image to markup fine detail, and the ability to map multi-spectral images into RGB space in a large number of ways. A key point to note is that we do not require every pixel in an image to be classified by the analyst. In fact typically the analyst will only markup small portions of a large image, illustrating the various different contexts in which the desired feature appears, and similarly a variety of contexts in which the feature does not appear. Only the regions specifically marked up will be used as training data. The user can further restrict the amount of training data by drawing rectangular bounding boxes around selected regions of interest. Using Aladdin several different images can be marked up with the same feature, and these will all be used as training data. Figure 1 shows a screenshot of the user interface. Similar interfaces have been used to create training data for evolutionary algorithms by Bersano.[6]

2.2. The Learning System

At the heart of Genie is a genetic programming system based on a linear chromosome (see Banzhaf et al.[7] for a good introduction to GP with an emphasis on linear genomes). The GP system manipulates image processing programs that take the raw pixel data planes and transform them into a set of *feature planes.* This set of feature planes is in effect just a multi-spectral image of the same width and height as the input image, but perhaps having a different number of planes, and derived from the original image via a certain sequence of image processing operations. Genie then applies a conventional supervised classification algorithm to the feature planes to produce a final output image plane, which specifies for each pixel in the image, whether that feature is there or not. A fitness for the image processing program just examined is then calculated by comparing this output image with the training data in a manner described below. Figure 2 illustrates this hybrid scheme.

2.2.1. Genetic Programming Details

Genie uses a fixed-length linear chromosome, rather than the more conventional tree-based representation typically used in GP, and we use standard one-point crossover, rather than some form of more sophisticated sub-tree or sub-graph crossover. This choice is motivated partly by a desire to produce code similar to that which a human might produce (the individual elements in our genome actually correspond directly to lines of code written in the


Aladdin (/nh/kitchenaid/projects/daps/rfip/gtf/IKONOS/los_alamos/urban1.roads1.gtf
File
View
Server
Options
Genie
Help
Aladdin Reference Display (/n/projects/dap

Brush Size: 10

Paint Type: True

☑ Overlay Training Data

□ Overlay Results Data

☑ Show background

Feature:	🟩
Non-Feature:	🟥
False Positive:	🟨
False Negative:	🟥

Detection Rate:
False Alarm Rate:
Fitness:

Figure 1. The Aladdin interface. Training data is painted onto a grayscale version of an image as a colored overlay. In this example, roads inside a user-specified bounding box have been marked up in green to indicate that this feature is to be found, while the surrounding area has been marked up in red. This distinction will not be obvious if you're reading a black and white version of this paper!

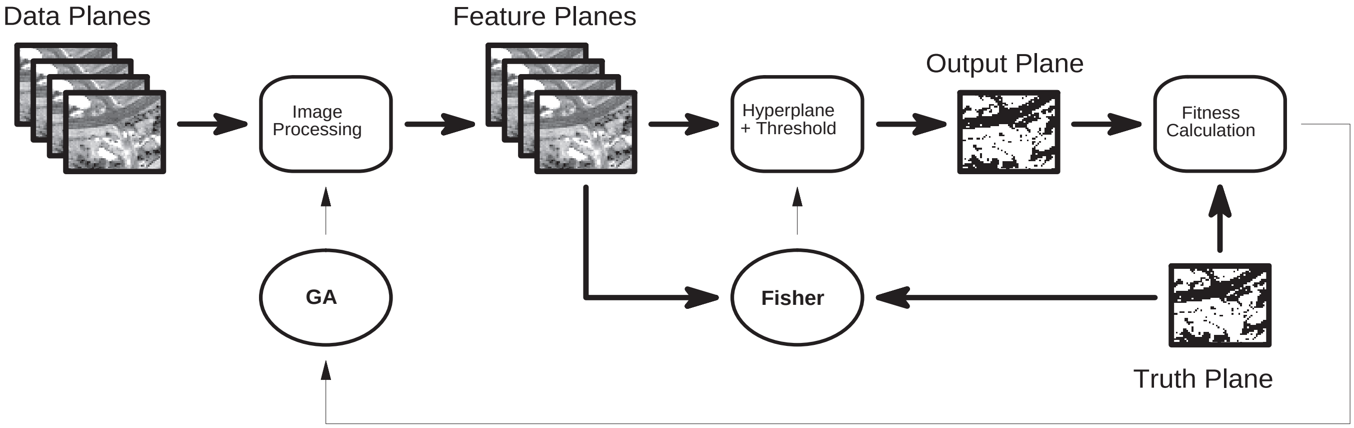


Figure 2. The overall structure of Genie. Raw data planes are transformed into a set of feature planes by an image processing program that is evolved by the GA. A discriminating hyperplane is then applied to those feature planes to produce a final classification. The hyperplane is found using a combination of Fisher discriminant and a threshold search. The output classification is compared with the user-supplied truth to obtain a fitness score, which is passed back to the GA.

ADDS,rD1,wS1,0.2	NDI,rD3,rS1,wS1	OPCL,rS1,wS2	SQRT,rS1,wS1	CLIP_HI,rS2,wS2,0.1

```
Scratch1 <= ADDS(Data1, 0.2)
Scratch1 <= NDI(Data3, Scratch1)
Scratch2 <= OPCL(Scratch1)
Scratch1 <= SQRT(Scratch1)
Scratch2 <= CLIP_HI(Scratch2, 0.1)
```

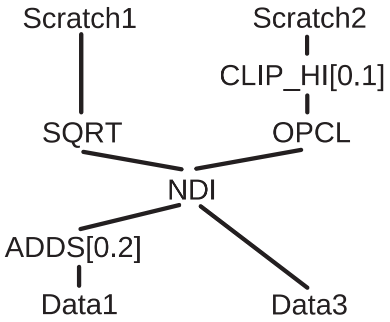


Figure 3. Three equivalent views of a chromosome. At the top is the raw linear genome representation that we use. Lower left shows a 'lines-of-code' version of the genome. This is the form closest to what gets executed. Lower right is a graph representation of the same genome. Note that the genome representation allows more complicated graphs than just simple trees.

data processing language IDL), and partly by the observation that it seems to work well. It is interesting to note that several others in the genetic programming community, e.g. Francone et al.,[8] have claimed significantly better performance with simple 'homologous' crossover on fixed length strings, compared with more flexible crossover schemes.

A single chromosome is made up of a string of genes, each one of which corresponds to a particular image processing operation. Each gene has one or more inputs, and one or more outputs. An input can be taken from any one data plane in the original image (there are as many data planes as there are spectral channels), or from any one 'scratch plane'. Scratch planes are temporary holding places where a single image plane can be held. The gene performs some image processing operation on its inputs and produces one or more planes of output data. Each of these planes is written to a different scratch plane. The whole chromosome is evaluated by starting with the gene at the left end, and sequentially stepping through the genes in order, one-by-one. It is a requirement when chromosomes are created that no gene is allowed to read from a scratch plane that has not been written to by at least one gene to the left of it. In addition, in these experiments, we impose a requirement that all scratch planes must be written to at least once during a chromosome's execution. The feature planes which are passed on to the backend classifier, are specified by the user as a subset of the scratch and data planes. Figure 3 shows how our genome representation is translated into an image processing pipeline.

Note that it is quite possible that a gene will write its output to a scratch plane, which is then overwritten by another gene before it is ever used. In this case that gene is irrelevant, as are any genes that write data to scratch planes that are only read by irrelevant genes before being overwritten. Hence, although the chromosome length is fixed, the effective program length can vary significantly. In Genie, we perform an efficient graph analysis of the chromosome and determine which genes are irrelevant. Those genes are kept in the chromosome, but for efficiency, are never actually executed.

Each gene corresponds to a different image processing operation, but the details of that operation can be influenced by gene parameters. Different genes have different numbers of parameters, and each parameter is associated with a fixed set of attributes that determine such things as what range it is randomly initialized within when that gene is first created, what range of values it can possibly take, and how it is affected by mutation (see below). We have three kinds of parameters: (*i*) float parameters are initialized to a random floating point number in the initialization range, and are mutated by a floating point offset that is Gaussian distributed with a standard deviation given by that parameter's *delta* attribute; (*ii*) integer parameters are initialized to a random integer in the initialization range, and are mutated by an integer offset that is uniformly distributed in a range given by plus or minus the *delta* attribute; (*iii*) symbolic parameters are like integers, but when mutated are simply re-initialized randomly. In general, genes attempt to produce output that is roughly on the order of the same scale as their input. A variable called `dataScale` is set to the range of values found in the training data, and genes that naturally have an output that is of a different range to their input, rescale their output to be of the order of `dataScale`. Table 1 lists all the genes used in our runs.

2.2.2. Conventional Classification Details

The image processing program specified by the genome transforms raw data planes into a set of feature planes. Typically, we take the scratch planes to be the feature planes. A conventional classification algorithm is then used to produce the final output classification, on which the genome fitness is based. The classifier uses the training data provided by the analyst, and attempts to find the best classifier it can that reproduces that training data. Since every genome evaluation requires a pass of the backend classifier, we require that this classifier run very quickly. As a result we use the relatively simple *Fisher linear discriminant.* See, for example, Bishop[9] for a detailed description of this algorithm, but put simply the Fisher discriminant is an optimal (in some sense) linear discriminant in feature space that takes into account the means of the true and false pixel clusters as well as their covariance matrices. The computational requirement for the Fisher discriminant is $\mathcal{O}(n^2)$ in the number of pixels,[*] and $\mathcal{O}(n^3)$ in the number of feature planes. In fact the Fisher discriminant only gives the normal to the discriminant hyperplane — we then do a line search ($\mathcal{O}(n)$ in the number of pixels) to find the optimal threshold along that direction. The output from the classifier is a single image plane containing 1's and 0's corresponding to true and false.

2.2.3. Fitness Calculation

Our fitness measure assigns equal importance to getting all the pixels marked as true correct, and getting all the pixels marked as false correct. The *detection rate* R_d is defined as the fraction of pixels marked as true that the classifier got correct, and the *false alarm rate* R_f is defined as the percentage of pixels marked as false that the classifier got wrong. The fitness F is then defined as:

$$F = 500 \times (R_d + (1 - R_f))$$

which gives a number between 0 (bad) and 1000 (perfect). In fact due to the operation of the Fisher discriminant, it is impossible to get a score less than 500 on the training data. The fitness score is then assigned to the genome that was just used.

When the initial population of chromosomes has been evaluated, conventional tournament selection is used to select parents for the next generation. Two parents are selected at a time. Single point crossover is performed with probability P_c. We perform up to N_g single-gene mutations on each individual, where N_g is the number of genes in the genome. Each of these mutations has a $\frac{P_m}{N_g}$ probability of happening, but if we didn't perform crossover, then at least one mutation always happens. Three kinds of mutation can happen: replacement of the gene with a new random gene, mutation of a single parameter of a gene, and mutation of a single input or output plane specification. These mutations happen in the ratio 2:1:1. Elitism is employed so that the single best individual is copied into the next generation. A check is performed on offspring chromosomes to see that they do not contain genes that read from uninitialized scratch planes, and to make sure that all the scratch planes are used at least once. Offspring that fail this check are discarded. Breeding continues until a new population of equal size to the original population has been created, and the cycle repeats.

2.3. Examining Results

At the end of a run, the best classifier genome is saved, along with the coefficients describing the backend classifier hyperplane. The analyst can then run this algorithm over other images and see whether the classifier is doing a good job. ALADDIN is again used as a tool for overlaying classification results on top of an image. If additional marked up images are available, then ALADDIN can overlay results data on top of training data to show where the evolved algorithm disagrees. On the basis of this the analyst might decide to mark up additional training data, and begin a new training run, perhaps seeding the run with the best of the previous run. This iterative process continues until the analyst is satisfied that a robust classifier has been developed. We are currently also examining other ways of combining together classifiers evolved during different runs using techniques such as *boosting.*[10]

[*] Note that only pixels in the images that have been assigned true or false values are considered.

Gene Abbreviation	Image Processing Operation	Inputs/ Outputs/ Params	Notes
ADDP	Add planes	2/1/0	Basic mathematical operations. ADDS adds a scalar, which may be negative, to its input. DIFF is like SUBP but outputs the absolute values. NDI is like SUBP, but divides the result by the sum of its two inputs. MULTS scales its input by a scalar, which by default is positive. LINSCL is like MULTS but takes an extra param which is added onto the scaled input. LINCOMB outputs a linear combination of its two inputs, in proportion specified by its one parameter, which takes a value between 0 and 1.
ADDS	Add scalar	1/1/1	
SUBP	Subtract planes	2/1/0	
DIFF	Absolute difference	2/1/0	
NDI	Normalized difference	2/1/0	
MULTS	Multiply by scalar	1/1/1	
NEG	Negate plane	1/1/0	
MULTP	Multiply planes	2/1/0	
SQRT	Square root	1/1/0	
SQR	Square	1/1/0	
LINSCL	Linear scale	1/1/2	
LINCOMB	Linear combination	2/1/1	
MIN	Minimum	2/1/0	Logical operations. MIN and MAX perform pixelwise minimum and maximum, equivalent to AND and OR for binary input. IFLTE outputs its third input wherever the first input is less than its second input, and its fourth output elsewhere.
MAX	Maximum	2/1/0	
IFLTE	'If less than else'	4/1/0	
CLIP_HI	Clip high	1/1/1	Thresholding operations. CLIP_HI truncates any pixel values above a value set by its param. CLIP_LO does the converse. THRESH sets all values below its threshold param to 0, and all those above to `dataScale`.
CLIP_LO	Clip low	1/1/1	
THRESH	Threshold	1/1/1	
SAVAR	Spectral angle variance	2-16/1/2	Spectral angle operations. SAVAR and SADIST look at two circular neighborhood regions around each pixel, of size defined by their two params. SAVAR returns the difference between the variance of the spectral angles of the pixels in the two regions. SADIF returns the difference between the mean spectral angle of both regions. SADIST returns the spectral angle difference between each pixel and the vector defined by its params. SANORM normalizes the vector defined by its inputs to have a magnitude equal to `dataScale`.
SADIF	Spectral angle difference	2-16/1/2	
SADIST	Spectral angle distance	2-10/1/2-10	
SANORM	Normalize spectral vector	2-10/2-10/0	
R5R5	Lawe's texture measure	1/1/0	Neighborhood operations. In general, all these operations take a single plane as input and produce a single output plane. The output at each pixel is determined by looking at the pixel's neighborhood. R5R5, LAWB, LAWD, LAWF and LAWH are widely-used texture measures, developed by Lawe, that return zero if the neighborhood contains all the same value of pixel, and some other value otherwise, depending upon the distribution of pixel values. R5R5 is corresponds to Lawe's $R5^T \times R5$ 5×5 operator. The others are 3×3 operators, corresponding to Lawe's $S3^T \times L3$, $E3^T \times E3$, $L3^T \times S3$ and $S3^T \times S3$ operators repsectively. Most of the other operators are familiar image processing or morphological operators, whose description can be found in any good book on image processing. Most take a single param which gives the radius of a circular structuring element defining the neighborhood to which the operator is applied. ASF stands for 'Alternating Sequential Filter'. MB_EDGE takes an additional parameter defining a threshold for edge strength to be looked for. The single parameter for H_DOME and H_BASIN defines the pixel value offset used by these operators.
LAWB	Lawe's texture measure	1/1/0	
LAWD	Lawe's texture measure	1/1/0	
LAWF	Lawe's texture measure	1/1/0	
LAWH	Lawe's texture measure	1/1/0	
LAPLAC3	3x3 Laplacian	1/1/0	
LAPLAC5	5x5 Laplacian	1/1/0	
MORPH_LAPLAC	Morph. Laplacian	1/1/1	
ISO_GRAD	Isotropic gradient	1/1/0	
MEAN	Mean	1/1/1	
VARIANCE	Variance	1/1/1	
SKEWNESS	Skewness	1/1/1	
KURTOSIS	Kurtosis	1/1/1	
SKEW_COEFF	Skewness coefficient	1/1/1	
KURT_COEFF	Kurtosis coefficient	1/1/1	
SD	Standard deviation	1/1/1	
RANGE	Range	1/1/1	
MEDIAN	Median	1/1/1	
EROD	Erode	1/1/1	
DIL	Dilate	1/1/1	
OPEN	Open	1/1/1	
CLOS	Close	1/1/1	
OPCL	Open-close	1/1/1	
CLOP	Close-open	1/1/1	
ASF_CLOP	ASF Close-open	1/1/1	
ASF_OPCL	ASF Open-close	1/1/1	
POS_TH	Positive top hat	1/1/1	
NEG_TH	Negative top hat	1/1/1	
OPEN_REC	Open with reconstruction	1/1/1	
CLOSE_REC	Close with reconstruction	1/1/1	
H_DOME	H-dome	1/1/1	
H_BASIN	H-basin	1/1/1	
MB_EDGE	Canny edge detector	1/1/2	

Table 1. The genes used in Genie and what they do. Unfortunately, space precludes a complete description of the details of the more complex operators, but this table gives the general picture of what kinds of operators are in Genie.

3. EXPERIMENTS

Genie is a hybrid learning system. The aim of the experiments in this paper is to compare the performance of the hybrid system, with each of the individual learning components taken separately. In doing so we will also illustrate Genie finding successful solutions to two interesting feature classification problems.

We used data from the recently launched IKONOS satellite.[11] This satellite produces commercially available imagery of the Earth at very high resolution. The raw data product consists of panchromatic images at 1 m resolution, and 4-color (blue, green, red and near-IR) images at 4 m resolution. We used 4-color imagery that has been 'sharpened' to 1 m resolution using the panchromatic image — the intensity of the high-resolution panchromatic image is used for each pixel, with chromaticity provided by the low-resolution color image. The images used in these experiments consist of 1000×1000 tiles cut from a large 10928×10928 IKONOS image of Los Alamos County in New Mexico, USA. The IKONOS data provides particularly interesting tasks for Genie since the high resolution shows up considerable texture, meaning that a simple color-based classifier often performs badly.

Genie was tested on its abilities to find two different kinds of features: (A) roads and large paved areas, and (B) regions of forest that suffered severe burn damage during the Cerro Grande forest fire of May 2000. Two 1000×1000 tiles were created for each feature type, and partially marked up with the appropriate features using Aladdin. Figure 4 shows the images and the associated truth markup. Images A1 and B1 were used as training data for Genie runs, and the resulting algorithms were then tested on images A2 and B2.

Four experiments were performed using both training sets:

1. **GA Only:** Genie was run with the backend disabled. The image processing part was run in the same way as described before, but then the contents of the first feature plane, thresholded at a value of 128 (half the pixel value range in the raw image) was used directly as the classification output.
2. **GA+Threshold:** As the GA Only run, but instead of the fixed threshold, an optimal threshold is found using a 1-D search of values in the first feature plane.
3. **GA+Fisher:** The full Genie system as described above, including the Fisher Discriminant backend and optimal thresholding.
4. **Fisher Only:** Disabling the evolutionary component this time, and simply passing the four input data planes through directly to the feature planes, before deriving the Fisher Discriminant and optimal threshold.

The idea behind these experiments is to examine the relative importance of the evolutionary and backend components of Genie and to demonstrate how they work together.

The following parameters were used in the GA: population size = 100, $N_g = 20$, number of scratch planes = 4, tournament size = 2, elitism = 1, $P_c = 0.9$ and $P_m = 0.5$. The feature planes were specified to be all the scratch planes. This set of parameters was chosen from experience, and has been found to work fairly well over many different runs of Genie. Each run was continued until 5000 functionally different chromosomes had been evaluated.[†] Each experiment was repeated 5 times with different random number seeds and the results are an average of the 5 runs.

4. RESULTS

Figure 5 shows the fitness of the best individual *vs.* number of functionally distinct chromosomes evaluated, for each of the three experiments involving an evolutionary component, and for each of the two training sets used. Table 2 summarizes the best final performance for all four experiments on both training and test sets. Note that these 'best' scores are an average of the best from 5 separate runs of each experiment.

The graphs in Figure 5 show that the GA+Fisher runs attain a higher fitness, faster (in terms of chromosome evaluations) than either the GA Only or the GA+Threshold runs. Table 2 shows the full Genie runs generally attaining a higher final level of performance than any of the other runs on both training and test data, although given the small sample size, the picture is not completely conclusive.

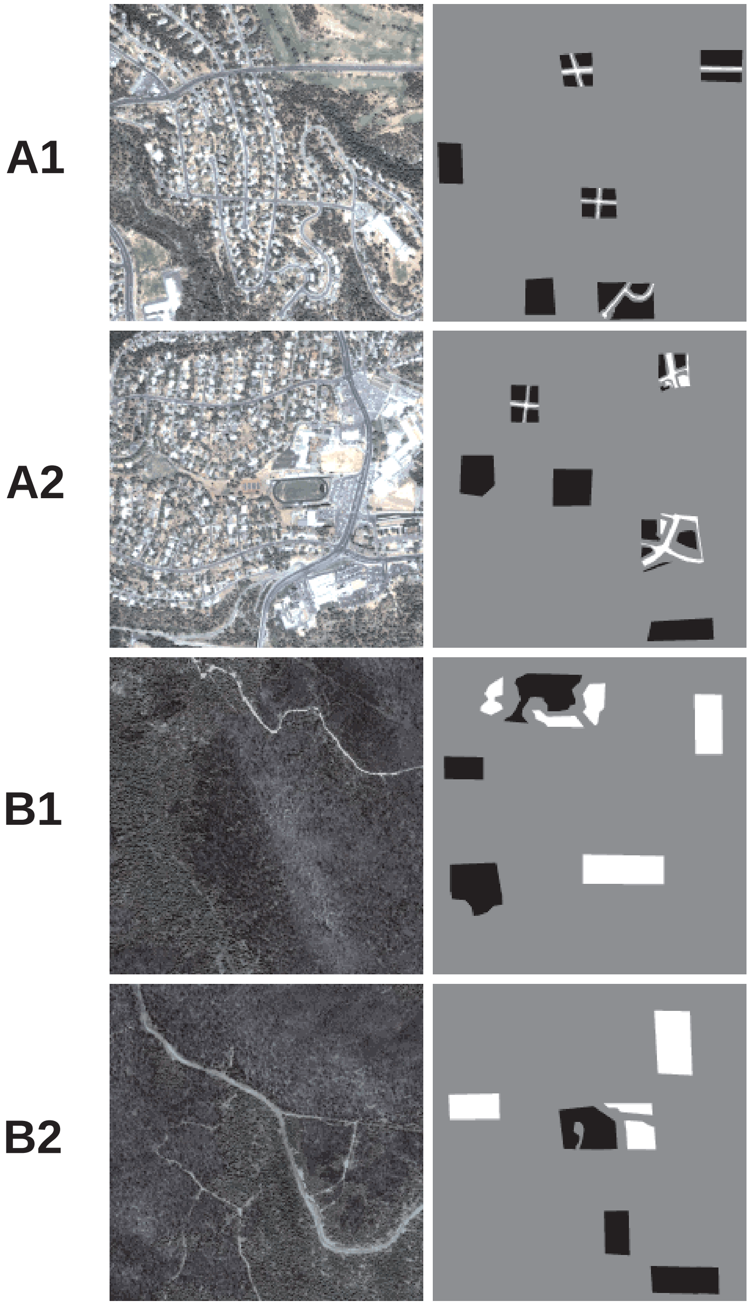


Figure 4. Training and test data used for Genie runs. The left hand column shows the 1000 × 1000 IKONOS tiles, with just the visible bands shown. The right hand column shows the training/test data that has been marked up for each tile. White indicates where the analyst has marked the feature of interest as being present, black indicates places that have been marked as not containing the feature, and gray indicates places that have not been marked. As can be seen, only small portions of each image have been marked up, which reduces the amount of image processing Genie must perform on each image.

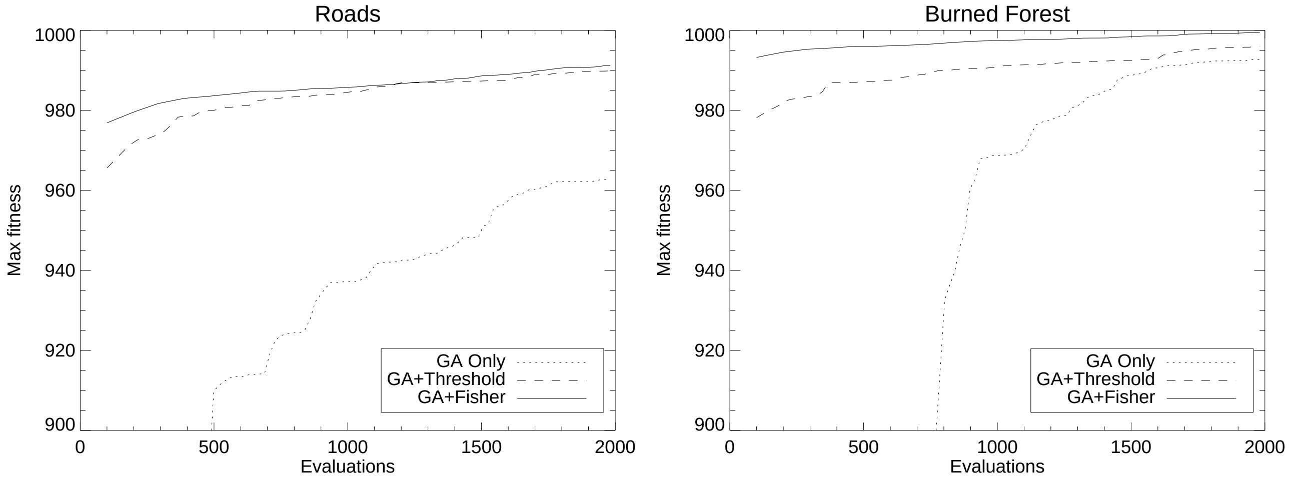


Figure 5. Plots of maximum fitness (averaged over 5 runs) *vs.* number of chromosome evaluations, for each of the three evolutionary experiments, and for each training set.

	Roads		Burned Forest	
	Training	Test	Training	Test
GA Only	965.2 ± 6.8	944.3 ± 9.5	991.6 ± 2.4	964.9 ± 1.6
GA+Threshold	987.6 ± 1.5	972.0 ± 2.6	997.5 ± 1.3	946.7 ± 7.8
GA+Fisher	988.7 ± 0.5	965.2 ± 4.5	999.0 ± 0.57	978.7 ± 6.8
Fisher Only	972.9	963.2	981.7	968.5

Table 2. Final maximum training and best test fitnesses (averaged over 5 runs) for all experiments and both training/test sets. The uncertainty in the fitness of the evolutionary runs gives the standard error of the mean of the 5 runs.[§]The Fisher Only experiment is deterministic and so has no uncertainty.

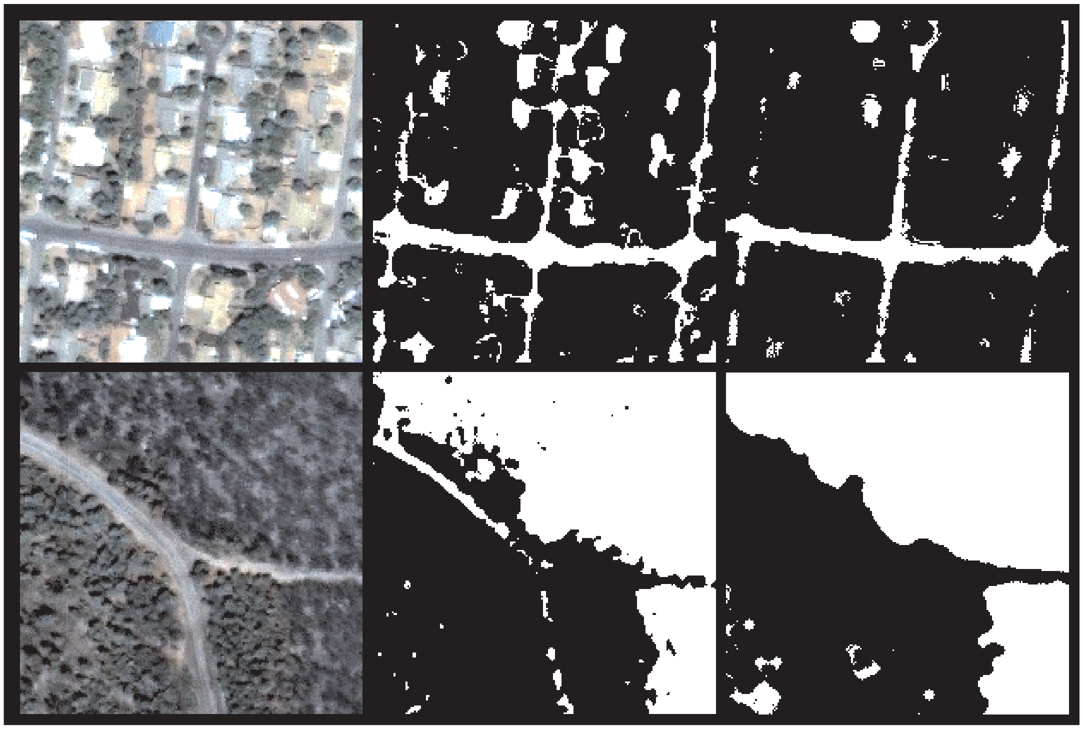


Figure 6. These images compare the classification output of the full Genie and Fisher Only classifiers on a small part of the IKONOS test images. On the left are small portions of the IKONOS images A2 (top) and B2 (bottom). The middle column shows the classification resulting from just applying the Fisher discriminant, while the right hand column shows the Genie classification. Neither algorithm does perfectly, but it is easy to see that the purely spectral Fisher Only classification is unable to distinguish between, say, roads and house roofs of similar color. The Genie classifier on the other hand is able to use spatial context to disambiguate the two largely.

It is interesting to see how classifications produced by the Fisher Only experiments compare subjectively to those produced by the full GA+Fisher runs. Figure 6 shows such a comparison.

Limited space precludes a detailed analysis of the feature planes that Genie came up with for these tasks, but one or two basic things are apparent. For the burned forest problem, many of the successful genomes produced a feature plane that was some sort of smoothed version of the near-IR channel. Living vegetation is generally brighter than the severely burned areas, and smoothing helps fill in the remaining textural variation. For the road finder the nature of the feature planes that Genie came up with is more obscure.

Final performance is not the whole issue of course. Since every single evaluation of a genome involves computing a Fisher discriminant, the Fisher Only experiments are thousands of times faster to run than the full Genie experiments: several seconds to get an answer compared to a small number of hours for Genie. The time taken to perform the image processing is of a similar order of time to that taken to compute the discriminant, and so the GA Only Runs take a similarly long time to run — although they are faster than the full Genie runs. Clearly there is a tradeoff between final performance and speed. In the typical scenarios that we work with, we can afford to take a few hours to get good final performance, but for some situations this may not be the case. An important point to note is that the GA operation is extremely parallelizable, and elsewhere[12] we have reported experiments on speeding up Genie using a cluster of workstations.

[†] We define 'functionally different' chromosomes as two chromosomes containing identical genes in the same order, after parsing to remove redundant genes.

[§] The standard error of the mean is defined as the sample standard deviation divided by the square root of the number of samples.

5. FURTHER WORK

In this paper we have described the GENIE system in considerable detail and we have presented results indicating that the combined GA plus conventional classifier system achieves higher performance than either the conventional classifier or the GA alone.

Further experiments need to be done to confirm this result conclusively. Another interesting question is whether some other kind of conventional classifier connected to the GA would work even better. We always have to bear in mind though that the more complicated the classifier, the slower each evaluation of the GA will become.

Our experience with GENIE has been that it can indeed automatically develop classification algorithms that are competitive with hand-designed classifier. We hope to continue to demonstrate this in future work, as well as working on improving the robustness and speed of the whole system.

ACKNOWLEDGMENTS

The GENIE system is the result of the combined efforts of a fairly large group of people at Los Alamos National Laboratory, including, in addition to the authors of this paper: Melanie Mitchell, Cody Young and Kevin Lacker.

REFERENCES

1. J. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, Springer, New York, 3rd ed., 1999.
2. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
3. J. Theiler, N. Harvey, S. Brumby, J. Szymanski, S. Alferink, S.Perkins, R. Porter, and J. Bloch, "Evolving retrieval algorithms with a genetic programming scheme," in *Proc. SPIE 3753*, pp. 416–425, 1999.
4. S. Brumby, J. Theiler, S. Perkins, N. Harvey, J. Szymanski, J. Bloch, and M. Mitchell, "Investigation of feature extraction by a genetic algorithm," in *Proc. SPIE 3812*, pp. 24–31, 1999.
5. N. Harvey, S. Perkins, S. Brumby, J. Theiler, R. Porter, A. Young, A. Varghese, J. Szymanski, and J. Bloch, "Finding golf courses: The ultra high tech approach," in *Real World Applications of Evolutionary Computing*, S. C. et al., ed., vol. 1803 of *Lecture Notes in Computer Science*, Springer, 2000.
6. T. Bersano-Begey, J. Daida, J. Vesecky, and F. Ludwig, "A Java collaborative interface for genetic programming applications: Image analysis for scientific inquiry," in *Proc. IEEE Int. Conf on Evolutionary Computation*, IEEE Press, Piscataway, NJ, USA, (Indianapolis), Apr. 1997.
7. W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming: An Introduction*, Morgan Kaufmann, San Francisco, CA, 1998.
8. F. Francone, M. Conrads, W. Banzhaf, and P. Nordin, "Homologous crossover in genetic programming," in *Proc. Genetic and Evolutionary Computation Conference (GECCO'99)*, B. et al., ed., vol. 2, pp. 1021–1026, Morgan Kaufmann, San Francisco, CA, 1999.
9. C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
10. Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Machine Learning: Proc. 13th Int. Conf.*, pp. 148–156, Morgan Kaufmann, 1996.
11. Space Imaging Inc. website: http://www.spaceimaging.com/.
12. N. Harvey, S. Brumby, S. Perkins, R. Porter, J. Theiler, A. Young, J. Szymanski, and J. Bloch, "Parallel evolution of image processing tools for multispectral imagery," in *Proc. SPIE 4132*, (San Diego, CA), July 2000.

Los Alamos
NATIONAL LABORATORY

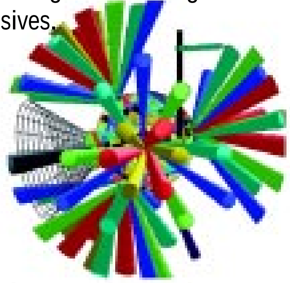
LOS ALAMOS PROFILE

Situated on more than 43 square miles in northern New Mexico, Los Alamos National Laboratory has more than 7700 regular full-time employees and an approximate annual budget of $1.5 billion. We are operated by the University of California for the National Nuclear Security Administration of the US Department of Energy.

Over our 58-year history, our primary mission has been to apply science and technology to problems of national security. At first our mission was dedicated principally to stockpile stewardship, but an ever-changing world has expanded our mission to cover other global threats, such as computer hacking and biological terrorism.

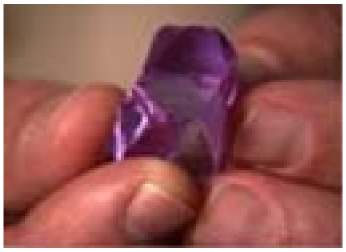
The Laboratory also conducts basic and applied research that addresses societal issues, such as developing alternative energy sources, designing the world's first functional quantum computer, and tracking down the most common ancestor of the HIV-1 strains responsible for AIDS. A vital facet to all our work is R&D collaboration with private industry. The following are five key R&D areas.

Maintaining Our National Security. The Laboratory continues to work on resolving nuclear weapons issues and on deciphering emerging technological challenges posed by the nuclear weapons stockpile. To accomplish this mission, the Laboratory applies an array of science and technology, from theoretical and computational physics to fabricating and testing explosives.



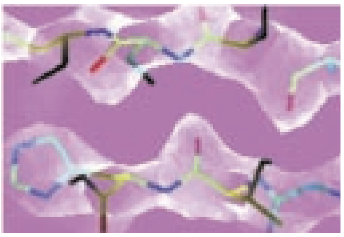
Scientists are developing hohlraums to achieve thermonuclear ignition, a process for the controlled release of great amounts of energy.

Developing Environmental Solutions. The Laboratory's expertise in this area spans a variety of environmental technologies, from waste minimization to environmental restoration and waste management. The principal goal of these and other programs is to maintain a safe and healthy environment for present and future generations.



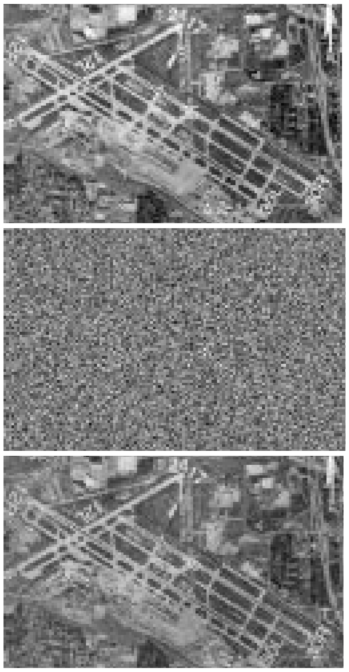
Radiation-tolerant materials like this could provide safe nuclear waste storage for thousands of years.

Understanding the Complexity of Life. At Los Alamos, scientists have united biological, physical, and computational sciences in an effort to better understand biological complexity. With such knowledge, we will develop technologies that address a number of critical issues, such as detecting, identifying, and defeating diseases and determining how genes function in the cell and the whole organism.



Researchers are using density contour maps to solve novel protein structures, many of which have medical applications.

Defeating Global Threats. Los Alamos researchers are developing technologies that defend the world against a number of international threats, such as the proliferation of nuclear weapons, chemical and biological agents, and information terrorism and computer infiltration.



Using a quantum cryptography process developed at Los Alamos, a sender can transmit an image (top) that is encrypted (middle) in such a way that only the intended receiver can decrypt it (bottom).

The Blue Mountain Supercomputing Platform helps researchers maintain the safety and reliability of the US nuclear stockpile.

Developing Supercomputers. In collaboration with industry, the Laboratory continues to develop faster and more sophisticated supercomputers to handle the extraordinarily complex calculations required to study the dynamics of nuclear weapons, global climate and ocean changes, or oil flow through underground rock.

Visit the LANL web site at http://www.lanl.gov/worldview/